AÑO III - NUM. 70

MICROHOBBY

SEMANAL

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES SINCLAIR

AÑO III - N.º 70

135 PTS.

Canarias 140 ptas.

MICROPANORAMA

11.000 PROGRAMAS PIRATAS REQUISADOS EN EL RASTRO DE MADRID

ESTRATEGIA

CREAMOS UNA AUTÉNTICA BATALLA DENTRO DE TU ORDENADOR

NUEVO

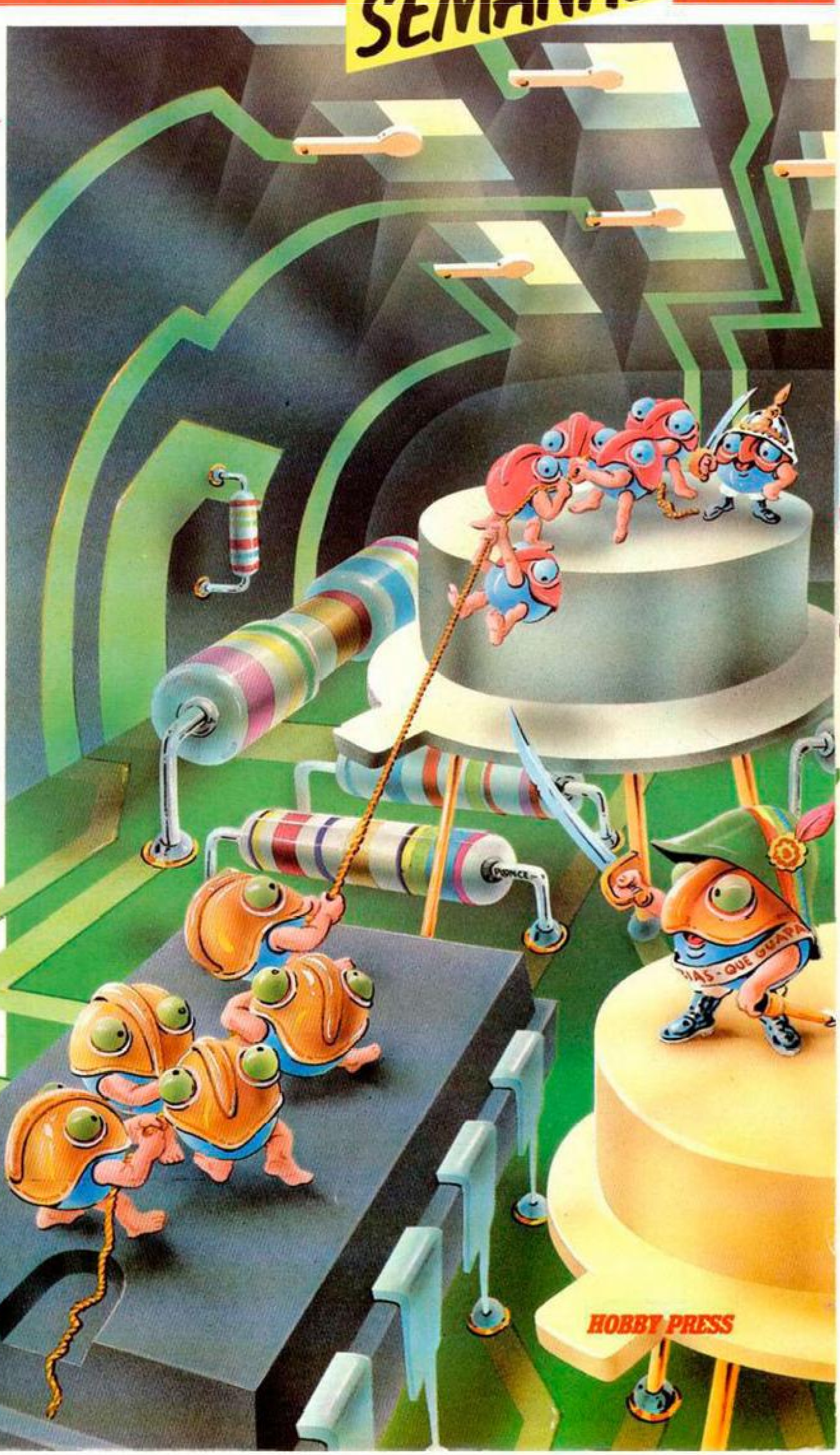
WINTER SPORT: EMOCIÓN SOBRE LA NIEVE

MICROMANIA

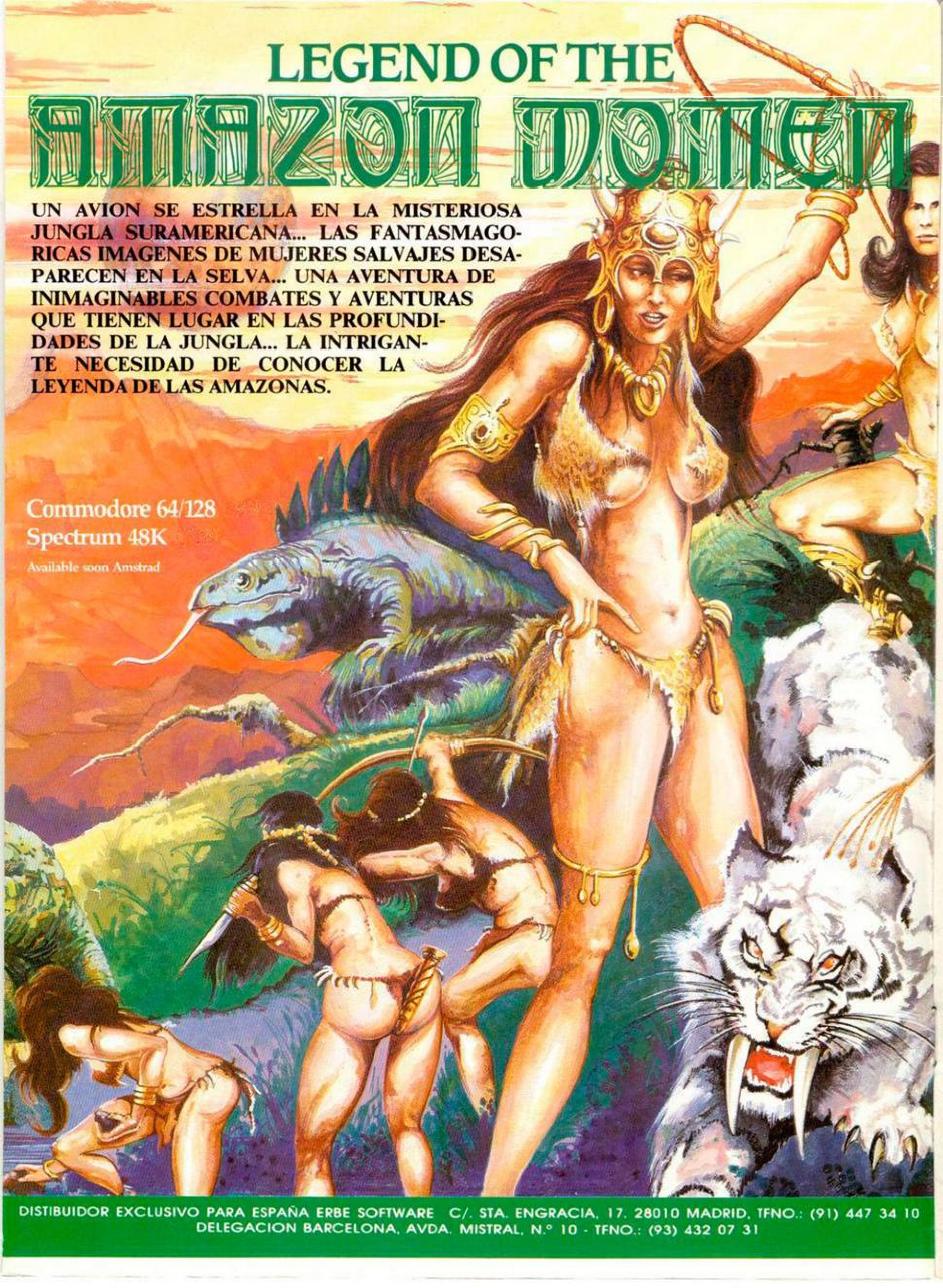
VIDAS INFINITAS PARA EL CAMELOT WARRIORS

RUTINAS DE UTILIDAD

HAZ QUE TU TECLADO FUNCIONE COMO UN JOYSTICK



HOBBY PRESS


LEGEND OF THE
AMAZON WOMEN
UN AVION SE ESTRELLA EN LA MISTERIOSA JUNGLA SURAMERICANA... LAS FANTASMAGORICAS IMAGENES DE MUJERES SALVAJES DESAPARECEN EN LA SELVA... UNA AVENTURA DE INIMAGINABLES COMBATES Y AVENTURAS QUE TIENEN LUGAR EN LAS PROFUNDIDADES DE LA JUNGLA... LA INTRIGANTE NECESIDAD DE CONOCER LA LEYENDA DE LAS AMAZONAS.
Commodore 64/128
Spectrum 48K
Available soon Amstrad
DISTIBUIDOR EXCLUSIVO PARA ESPAÑA ERBE SOFTWARE C/. STA. ENGRACIA, 17. 28010 MADRID, TFNO.: (91) 447 34 10
DELEGACION BARCELONA, AVDA. MISTRAL, N.° 10 - TFNO.: (93) 432 07 31

Director Editorial
José I. Gómez-Centurión

Director Ejecutivo
Domingo Gómez

Asesor Editorial
Gabriel Nieto

Redactora Jefe
Africa Pérez Tolosa

Diseño
Rosa María Capitel

Redacción
Amalio Gómez, Pedro Pérez, Jesús Alonso

Secretaria Redacción
Carmen Santamaría

Colaboradores
Primitivo de Francisco, Rafael Prades, Miguel Sepúlveda, Sergio Martínez y J. M. Lazo

Corresponsal en Londres
Alan Heap

Fotografía
Javier Martínez, Carlos Candel

Portada
José María Ponce

Dibujos
J. R. Ballesteros, A. Perera, F. L. Frontán, Pejo, J. M. López Moreno, J. Igual, J. A. Calvo, Lóriga, J. Olivares

Edita
HOBBY PRESS, S. A.

Presidente
María Andrino

Consejero Delegado
José I. Gómez-Centurión

Jefe de Publicidad
Marisa Esteban

Publicidad Barcelona
José Galán Cortés
Tels.: 303 10 22 - 313 71 76

Secretaria de Dirección
Marisa Cogorro

Suscripciones
M.ª Rosa González
M.ª del Mar Calzada

Redacción, Administración y Publicidad
La Granja, 39
Polígono Industrial de Alcobendas
Tel.: 654 32 11
Telex: 49480 HOPR

Dto. Circulación
Carlos Peropadre

Distribución
Coedis, S. A. Valencia, 245
Barcelona

Imprime
Rotedic, S. A. Ctra. de Irún, km 12,450 (MADRID)

Fotocomposición
Novocomp, S.A.
Nicolás Morales, 38-40

Fotomecánica
Grof
Ezequiel Solana, 16

Depósito Legal
M-36.598-1984

Representante para Argentina, Chile, Uruguay y Paraguay, Cía. Americana de Ediciones, S.R.L. Sud América 1.532. Tel.: 21 24 64. 1209 BUENOS AIRES (Argentina)

Solicitado control OJD

MICROHOBBY ESTA SEMANA

AÑO III. N.º 70. 18 al 24 de marzo de 1986.
135 ptas. (Incluido IVA)





Nuevo. «Winter Sports» vive toda la emoción de los deportes de invierno con este programa. (Pág. 12.)

PREMIADOS HOBBY-SUERTE

JORGE OLLOQUI ACHAELANDIA. C/ Gral. Alava, 11, 7.º Dcha. Vitoria (ALAVA).
Una cinta de programas (5.ª Cat.)
MARCOS E. MUÑOZ DE RIBERA. C/ S. Blas, 25. Alarcón (MADRID).
Una cinta de programas (5.ª Cat.)
ANTONIO J. GUTIERREZ SOLA. C/ Pacheco, 2. Ecija (SEVILLA).
Una cinta de programas (5.ª Cat.)
JOSE STGO. GARCIA. Virgen de la Antigua, 28, 8.º C. (SEVILLA).
Suscripción a MICROHOBBY semanal por un año (4.ª Cat.)
M.ª ANGELES PEREZ SANCHEZ. C/ Virgen de Africa, 36. (MADRID).
Una cinta de programas (5.ª Cat.)
CRISTOBAL MUÑOZ SANCHEZ. Avda. de Andalucía, 111. Ulpi (ALMERIA).
Suscripción a MICROHOBBY semanal por un año (4.ª Cat.)
JORGE GALLAR URBAN. C/ Villa de Lloret, 2, Esc-C. Blanes (GERONA).
Una cinta de programas (5.ª Cat.)
MARIANO SANTANA CLANYS. C/ Alfredo Marquerie, 47. (MADRID).
Una cinta de programas (5.ª Cat.)
M.ª PAZ MORADO ALVAREZ. C/ Ramón Patuel, 3, 6.º D. (MADRID).
Suscripción a MICROHOBBY semanal por un año (4.ª Cat.)
FRANCISCO PADILLA LORENZO. C/ Canarias, 16, 1.º Izqda. (MADRID).
Una cinta de programas (5.ª Cat.)
ALBERTO LLUPI JIMENEZ. C/ Zaramagos, 16, 3.º B. (MADRID).
Una cinta de programas (5.ª Cat.)
PABLO GOMEZ SANCHEZ. C/ Hernán Cortés, 24. Novelda (ALICANTE).
Una cinta de programas (5.ª Cat.)
JOSE M.ª ALVAREZ PAVIEL. C/ Pez Volador, 9. (MADRID).
Una cinta de programas (5.ª Cat.)
JESUS MORENO MONTEVERDE. Paseo de los Naranjos, 7. (HUELVA).
Una cinta de programas (5.ª Cat.)
ISMAEL PRIETO FERNANDEZ. C/ Naranjo de Bulnes, 14, 5.º A. Gijón (ASTURIAS).
Una cinta de programas (5.ª Cat.)
JULIO MONTOTO MENENDEZ. C/ Escultores, 11, 1.º Izqda. (MADRID).
Una cinta de programas (5.ª Cat.)
RAFAEL MARTIN SAEZ. C/ Cauce, 3, 4.º 1.ª. Alcobendas (MADRID).
Un Joystick con su Interface (3.ª Cat.)
ANTONIA ALAS BORRAJO. Avda. Arturo Arévalo, 2. Siete Aguas (VALENCIA).
Una cinta de programas (5.ª Cat.)
ANGEL MIGUEL DIAZ ROJAS. C/ Granada, 8. Miguelturra (CIUDAD REAL).
Una cinta de programas (5.ª Cat.)
MIGUEL ANGEL CRISOSTOMO. Arroyo Belincoso, 38. (MADRID).
Una cinta de programas (5.ª Cat.)

28 PIRATAS DETENIDOS EN EL RASTRO MADRILEÑO

En la mañana del pasado domingo 2 de marzo, los habituales vendedores de software pirata del Rastro madrileño se llevaron una desagradable sorpresa. En el momento álgido de las ventas y cuando se las prometían más felices, miembros del grupo 7.° de la Brigada Regional de la Policía Judicial procedieron pacíficamente a incautarles el «género» clandestino.



La magnitud de la operación puede apreciarse con facilidad si tenemos en cuenta que se intervinieron algo más de 11.000 programas, con un valor en el mercado de más de 22 millones de pesetas, lo que supone casi la quinta parte del total de las ventas legales de todo el año.

Los 28 detenidos fueron puestos a disposición Judicial, y más tarde en libertad. ANEXO, Asociación Española de Empresas Fabricantes de Soportes Lógicos, manifestó su intención de querellarse con cada uno de ellos por apropiación ilícita de los derechos de autor de estos programas, así como en lo referido a Propiedad Industrial de marcas, logotipos, anagramas y títulos.

Por otra parte, estos vendedores clandestinos, no tienen ningún tipo de permiso, ni licencia fiscal, ni cotizan a la Hacienda Pública, por lo que pueden ser procesados por fraude Fiscal.

LOS PROGRAMAS DE ORDENADOR SE INCLUYEN EN EL PROYECTO DE LEY DE PROPIEDAD INTELECTUAL

El Congreso de los Diputados ha publicado en su Boletín Oficial correspondiente al día 1 de febrero de 1986, un proyecto de Ley de la Propiedad Intelectual, en el cual se contempla, por primera vez, un título compuesto por seis artículos, dedicados exclusivamente a los programas de ordenador.

Según se desprende de dicho Boletín, este nuevo Proyecto de Ley de Propiedad Intelectual surge por la necesidad de renovar la anterior ley promulgada el 10 de enero de 1879, la cual, a pesar de que contribuyó en su momento a la formación de la propia tradición jurídica en la materia, no podía prever las profundas transformaciones sociales y las consecuencias del desarrollo de los medios de comunicación en las obras de creación, que han permitido, paralelamente el acceso de los ciudadanos a la cultura, nuevas modalidades de fraude a los derechos de la propiedad intelectual.

Uno de los objetivos que se persigue con este Proyecto de Ley, es el delimitar los derechos y obligaciones de los autores y explotadores de obras tecnológicamente avanzadas, como son las audiovisuales o los programas de ordenador, cuya definición también aparece en la presente ley, especificándose que «se entenderá por programa de ordenador toda secuencia de instrucciones o declaraciones destinadas a ser utilizadas, directa o indirectamente, en un sistema informático para obtener un resultado determinado, cualesquiera que fuese su forma de expresión o fijación».

El resto de los artículos están dedicados a delimitar más específicamente el campo de los derechos de los autores, y todos ellos serán tema de un informe más exhaustivo en un próximo número de MICROHOBBY.

En definitiva, este Proyecto de Ley supone un paso muy importante para todos los programadores y público en general relacionado con la informática, quienes podrán disfrutar finalmente de una legislación actualizada y moderna en la que se contemplan todos los aspectos relacionados con los derechos y obligaciones que se derivan de sus propias obras de creación.

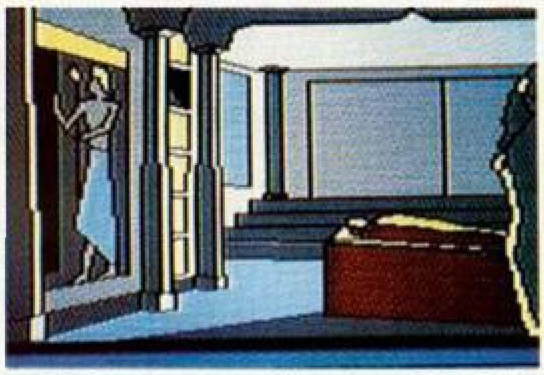
Por la compañía ODE

«MACBETH» HECHO AVENTURA

Macbeth, la maravillosa novela del no menos genial escritor inglés William Shakespeare, ha sido llevada al ordenador en forma de aventura. Por el momento, tan sólo estará disponible para el C64.

Este ambicioso proyecto ha sido llevado a cabo por Oxford Digital Enterprises, una pequeña y próspera casa de software fundada por Dr. Davis Pringle y Dr. Gareth Blower, ambos doctores en física nuclear cuyo equipo se completa actualmente con un ingeniero y un matemático, además de un selecto grupo de artistas gráficos y escritores.





Si Shakespeare levantara la cabeza, posiblemente desaprobaría la idea; sin embargo, Macbeth supone un gran logro en el mundo de las aventuras por ordenador. Está compuesta por cuatro partes distintas, cuatro sesiones con el psiquiatra del rey, en las que tendremos que enfrentarnos a invasiones, ganar, defender y perder coronas, maleficios de brujas, etc... todo ello realizado con un alto nivel técnico y de calidad gráfica.

ODE está, además, trabajando en un sofisticado programa de simulación de aterrizaje del Apolo sobre la superficie lunar, que llevará el nombre de «Reach for the moon», será lanzado al mercado el próximo verano por Rainbird Software.

PAQUETES ALSIPACK EN DISKETTE PARA EL QL

Alsi Comercial acaba de lanzar diez nuevos paquetes Alsipack, disponibles sólo en diskette, en los que se incluyen diferentes combinaciones de programas de gestión para el ordenador Sinclair QL.

Estos programas son:

COMERCIAL 6: Facturación, control automático de stocks y fichero de direcciones Mailing.

ALSISTOCKS: Control de stocks.

CAMBIALSI: Letras de cambio.

ALSIFIN: Simulador financiero, matemática financiera y estadística financiera.

ALSICOPY: Copiador de ficheros y programas.

FINCASALSI: Administración de fincas.

Estos Alispack contienen de tres a cuatro de estos programas y sus precios varían desde 25.000 ptas. del paquete que contiene Alsimail-Cambialsi + Alsicopy, hasta las 70.000 del que incluye Comercial 6 + Alsicont + Cambialsi + Alsifin.



ALSICONT: Contabilidad (diario, balances, extractos de cuentas, regularización y cierre).

ALSIMAIL: Fichero de direcciones Mailing e Impresión de recibos.

En Lérida

I SEMANA DE LA INFORMATICA

Los alumnos del último curso de la Escuela de Informática de Formación Profesional de Lérida, están llevando a cabo en estos días un simposio, con carácter anual, con el que se pretende paliar la escasa existencia de actos divulgativos en los temas referentes a la actualidad de la informática en nuestro país.

Esta primera edición tendrá lugar durante los días 17 al 21 del presente mes de marzo y consistirá en un monográfico acerca del tema de la Inteligencia Artificial.

Los principales actos previstos consistirán en un ciclo de conferencias, actividades audiovisuales, demostraciones prácticas y cursos elementales de formación sobre la I.A.

La realización de estas sesiones tendrá lugar en dos escenarios diferentes: el ciclo de conferencias y las actividades audiovisuales se celebrarán en el salón de actos de «La Caixa», mientras que las demostraciones prácticas y los cursos lo harán en la I.P.F.P. La Caparrella.

Estos cursos versarán sobre los lenguajes de la Inteligencia Artificial (LISP y Sistemas Expertos) así como de otros temas como Gestores de Bases de Datos y Hojas Electrónicas.

AQUI LONDRES

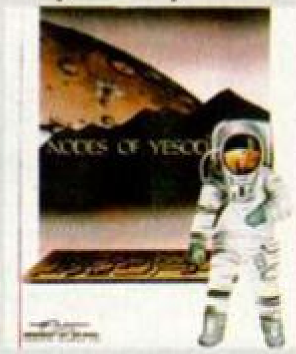


Según los últimos sondeos, parece ser que el **AMIGA** de **COMMODORE** va a ser lanzado inminentemente en Europa, y lo hará con una versión superior al 256 K que actualmente se encuentra disponible en USA, donde se vende con una unidad de disco incorporada. La versión europea estará prevista, según todas las previsiones, de una memoria de 512 K y dos unidades de disco, ambas, la memoria y la unidad de disco extras, son opcionales y se venden separadamente.

Sólo un pequeño inconveniente ensombrece el lanzamiento: su precio, ya que se cotizará a algo más de 1.500 £.

British Telecom sigue ampliando su departamento de software con gran rapidez. Hasta ahora, la poseedora de Firebird, Rainbird y Beyond acaba de adquirir **Odín**, la compañía que llegó a la fama con juegos como «Nodes of yesod», «Arc of yesod» y «Robin





of the wood». Aunque Odín permanecerá como una compañía separada, British Telecom tendrá la exclusiva mundial de los derechos de publicación y distribución de todos los títulos por salir.

El Ministerio de Comercio e Industria británico va a gastarse 1 millón de £ en **MODEM'S** para los colegios del Reino Unido. Un MODEM es un aparato que conecta un micro a otro, situado a cientos de km, a través de líneas telefónicas. También posibilita enlazar con grandes ordenadores centrales y por consiguiente, con enormes bases de datos que contienen gran variedad de información.

De nuestro corresponsal en Londres
ALAN HEAP

TRUCOS

NUMEROS PRIMOS

Joan Adam Bernad nos demuestra con el ejemplo que os ofrecemos a continuación, cómo descomponer un número en factores primos.

```
  10 DEF FN A(M,N)=M-(INT (M/N))
*N
  20 INPUT B: IF B<2 THEN GO TO
20
  25 LET A=2: PRINT B;"=";
  30 IF A=B THEN PRINT A: PRINT
#1;"Scroll?": PAUSE 0: GO TO 20
  40 IF FN A(B,A)=0 THEN LET B=B
/A: PRINT A;"*";: GO TO 30
  50 LET A=A+1: GO TO 30
```

Joan nos envía también un segundo truco para simular la operación «mod». Con esta función podemos averiguar el resto de una división.

```
  10 DEF FN A(M,N)=M-(INT  (M/N)
)*N
  20 CLS : INPUT "DIVIDENDO ";M
  30 INPUT "DIVISOR ";N
  40 PRINT AT 8,0;M;"/";N
  50 PRINT AT 10,0;"EL RESTO DE
 LA DIVISION ES ";FN A(M,N)
  60 PAUSE 0: GO TO 20
```

LA ALTERNATIVA

Francess Artigues nos ha mandado un truco alternativo al programa «Cirujano» que publicábamos en la sección de Utilidades del número 54 de la revista con el que se remediaba la mala carga de algunos programas «indomables».

— Hacer un miniprograma que sea número línea erróneo REM y salvarlo como SAVE «BISTURI».

— Cargar el programa roto.

— Hacer MERGE «BISTURI».

Y ya está resuelto el problema.

EFECTOS DE «PAPER» Y «BORDER»

Este truco de Alberto Rojo nos proporciona un sonido un tanto raro y un efecto de «PAPER» y «BORDER» fascinante. Probarlo y lo comprobaréis.

```
  10 CLEAR 29999: FOR l=30000 TO
 30028: READ a: POKE l,a: NEXT l
  20 DATA 14,255,6,29,33,0,91,62
,239,113,0,211,254,43,61,194,57,
117,5,120,194,55,117,13,121,194,
50,117,201
  30 RANDOMIZE USR 3e4
  40 STOP
```

MOVIMIENTO SIN INKEY$

Esta rutina de José Luis Vendrell permite que el Spectrum reconozca la tecla pulsada sin necesidad del comando «INKEY$».

Como todos sabemos, el



inconveniente de este comando es el poder pulsar solamente una tecla, con lo que en nuestros juegos no hay posibilidad de movimiento en ocho direcciones sin que ello resulte demasiado engorroso (ya que deberíamos controlar ocho teclas). La solución a este problema la encuentra José Luis en el comando «IN» ya que éste reconoce la tecla pulsada permitiendo, además, pulsar varias a la vez.

Si tecleamos este programa, lo veremos más fácilmente.

```
   1 LET a=1
   2 LET b=1
   3 LET c=IN 63486
   4 IF c=191 THEN LET c=190
   5 IF c=255 THEN LET c=254
  10 IF IN 64510=c THEN LET a=a-
1
  20 IF IN 57342=c THEN LET b=b+
1
  30 IF IN 65022=c THEN LET a=a+
1
  40 IF IN 61438=c THEN LET b=b-
1
  45 PRINT AT a+1,b;"   "
  50 PRINT AT a,b;" x "
  60 PRINT AT a-1,b;"   "
  70 GO TO 10
```

Existen en el mercado varias series de Spectrum, para los cuales la función «IN» es diferente.

En las versiones 2 y 3 B, el valor de esta función es 64 más que en la versión 3.

Pero para ver esto con más detalle basta hacer PRINT IN 63486; si es de versión 2 ó 3 B obtendremos el valor 255, mientras que en la otra versión nos dará 191.

¡MENUDO SUSTO!

J. Ismael Ripoll, nos recomienda este pequeño truco para «asustar» en determinadas partes de un programa.

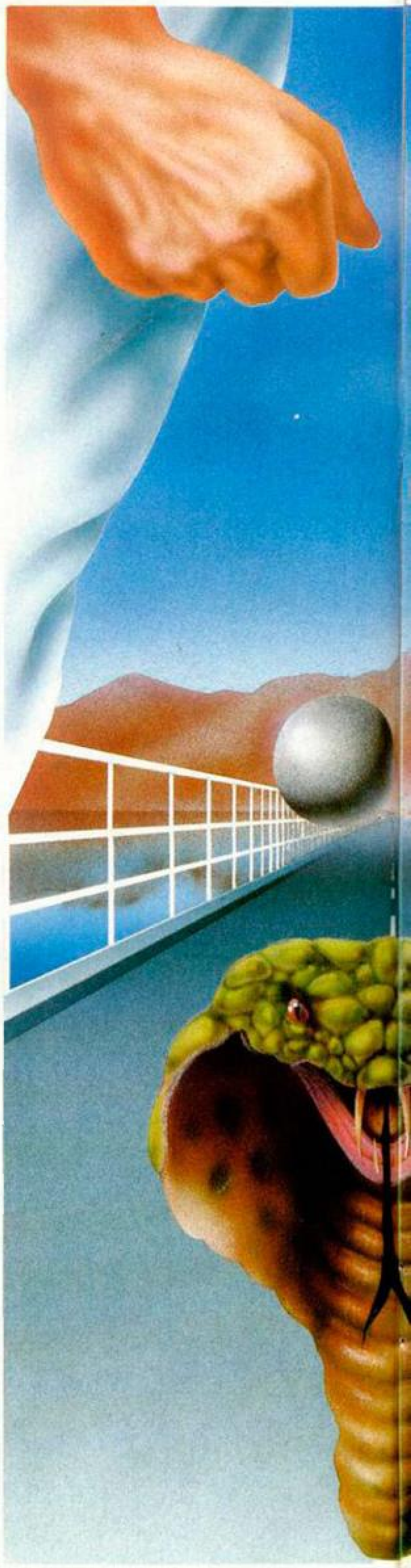
```
  10 RANDOMIZE USR 11146: RANDO
MIZE USR 3190: GO TO 10
```

Si queréis colores claros, habrá que ponerlo en marcha con CONTINUE, y si, por el contrario los queréis oscuros, con RUN.

RIO KARATE

José Antonio AVILA

Spectrum 48 K



Estamos al límite de nuestras posibilidades y una última prueba viene a empeorar la situación. Exhaustos y hambrientos, hemos de cruzar el río que nos separa de la civilización y... de nuestra salvación.

Con el enemigo pisándonos los talones no debemos dudar en atravesar el misterioso puente que une las dos orillas del denominado río Karate (pronto comprenderemos el porqué de esta denominación). Cautelosos, llegamos hasta la mitad del puente cuando un personaje nos corta el paso. Es un experto en artes marciales y nos vemos obligados a demostrar nuestra habilidad con el karate si no queremos sucumbir ante el enemigo. Tras el primer derrotado, aparece otro, pero además, algo se acerca deslizándose por el suelo... Serpientes venenosas y bolas rodantes se abalanzan sobre nosotros, esquivarlas será nuestra única salvación. Esto y derrotar a los karatekas que no cesan en su empeño de aniquilarnos... Las teclas de movimiento son:

Q-W, para la derecha y para la izquierda.

I, para dar una patada.

O, para saltar.

P, para agacharse.

¿Te interesa, un truco para salir con vida de la situación? Estamos seguros de que sí. Pues bien, sigue estos consejos:

— Si te viene un hombre, no hay más remedio que darle una patada y echarlo fuera del puente.

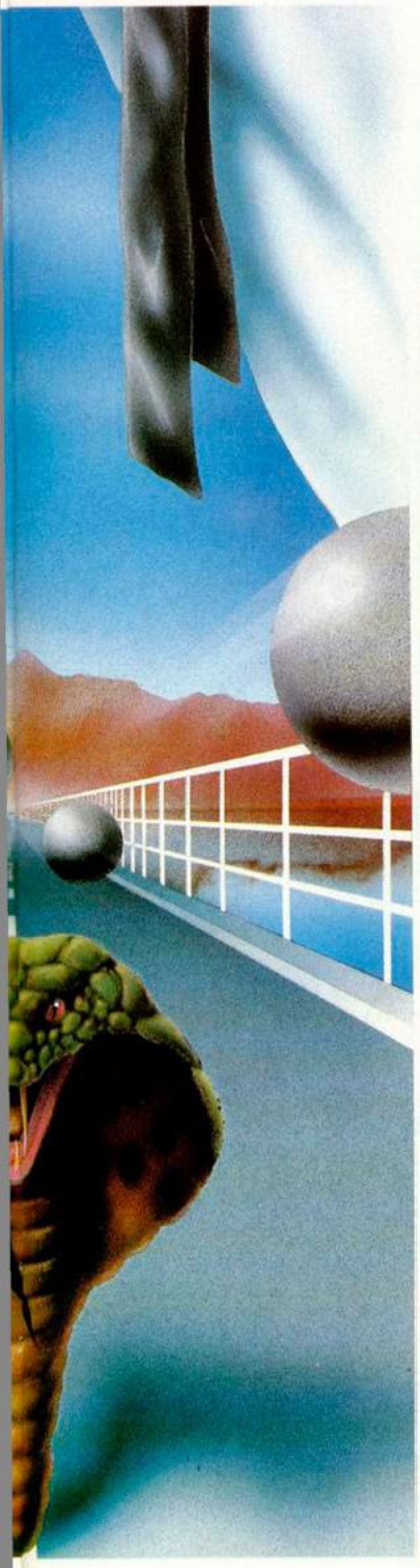
— Si te viene una serpiente, hay que saltarla.

— Si lo que te viene es una bola baja, sáltala; si viene alta, hay dos opciones, o te agachas o la das una patada.




```
   1 REM JOSE ANTONIO AVILA RUI"
   2 POKE 23658,8
   3 LET J=7: LET I=5: LET H=7:
LET mov=29: LET Y$="C": LET J$="
D": LET R$="D": LET S$="N": LET
A=2
   4 LET D$="N": LET PUN=0: LET
VID=5: LET E=100
   5 LET D=0: INK 7: PAPER 0: BO
RDER 0: CLS
   9 REM [illegible]
  10 FOR a=0 TO 20: FOR w=0 TO 7
  20 READ s: POKE USR CHR$ (144+
a)+w,s
  30 NEXT w
  40 NEXT a
  50 DATA 16,56,56,16,56,60,122,
121
  60 DATA 152,56,60,60,52,102,66
,99
  70 DATA 8,28,28,8,28,60,94,158
  80 DATA 28,28,60,60,44,102,66,
198
  90 DATA 2,7,7,2,7,233,195,126
 100 DATA 0,0,0,0,0,189,255,3
 110 DATA 0,0,0,0,0,19,31,0
 120 DATA 64,224,224,64,224,151,
195,126
 130 DATA 60,24,24,24,24,16,16,2
4
 140 DATA 0,0,0,0,0,200,248,0
 150 DATA 0,0,0,0,0,111,248,111
 160 DATA 0,0,0,0,0,189,255,192
 170 DATA 0,24,60,126,126,126,60
,24
 180 DATA 0,0,0,0,3,58,110,196
 190 DATA 222,243,219,19,127,252
,126,2
 200 DATA 123,207,219,48,126,63,
126,64
 210 DATA 255,36,66,129,129,66,3
6,255
 220 DATA 1,3,3,15,31,63,63,127
 230 DATA 0,128,192,240,248,252,
252,255
 240 DATA 0,0,0,0,0,246,31,246
 250 DATA 129,40,66,8,85,0,42,12
8
 299 REM PRESENTACION
 300 LET a$="[illegible]"
 310 LET b$="[illegible]"
 320 LET c$="[illegible]"
 330 LET e$="[illegible]"
 335 LET d$="[illegible]"
 350 LET f$="[illegible]"
 360 INK 3: PRINT AT 0,1;a$;AT 1
,1;b$;AT 2,1;c$: INK 9: PRINT AT
 4,0;e$: FOR a=5 TO 16: PRINT AT
 a,0;f$: NEXT a: PRINT AT 17,0;d
$
 380 PLOT 6,137: DRAW 243,0: PLO
T 6,137: DRAW 0,-100: PLOT 6,37
 DRAW 243,0: DRAW 0,100
 450 FOR a=5 TO 16: OVER 0: FOR
b=1 TO 30: INK 5: BEEP 0.001,40
 PRINT AT a,b;"█": NEXT b: NEXT
a
 460 BEEP 0.05,20: PRINT AT 20,1
;"PULSA UNA TECLA PARA EMPEZAR"
 470 PAUSE 0
 475 BEEP 0.05,20: PRINT AT 20,1
;"
```

JAVIER IGUAL



```
480 PRINT PAPER 4; INK 1;AT 19,17;" PUNTOS : ";PUN
482 PRINT PAPER 4; INK 1;AT 19,1;"ENERGIA : ";E
483 PRINT PAPER 4; INK 1;AT 21,1;"VIDAS : ";VID
499 REM PAISAJE
500 INK 0: PAPER 5: DIM T$(3,404)
505 LET T$(1)=""
510 LET T$(2)=""
511 LET X$=""
512 LET U$="S": FOR U=1 TO 9: LET U$=U$+U$: NEXT U: LET U$=U$( TO 404)
515 LET T$(3)=X$+U$+X$+"█"
530 LET g=1
535 LET G=G+1
537 IF g>=375 THEN GO TO 9700
540 PRINT INK 3;AT 16,1;T$(1,g TO g+29);AT 15,1;T$(2,g TO g+29): PRINT INK 0;AT 14,1;T$(3,g TO g+29)
550 GO TO 930
800 REM *MOVIMIENTO ATRAS**
802 GO SUB 9000
805 LET MOV=MOV-1: GO SUB 2000
810 INK 2: IF INKEY$="Q" THEN LET M=M-1
820 IF M<=2 THEN LET M=M+1
830 IF INKEY$="I" THEN GO SUB 1100
840 IF INKEY$="U" THEN GO SUB 900
850 IF INKEY$="O" THEN LET Z$="Q": GO SUB 1200
855 IF INKEY$="P" THEN LET X$="Q": GO SUB 1300
860 PRINT AT 12,M;"C";AT 13,M;"D";AT 12,M+1;" ";AT 13,M+1;" "
870 GO TO 800
900 REM MOVIMIENTO DELANTE
901 GO SUB 9000
902 LET MOV=MOV-1: GO SUB 2000
903 PRINT AT 19,26;PUN
910 INK 2: IF INKEY$="U" THEN LET M=M+1
920 IF INKEY$="Q" THEN GO TO 830
930 IF INKEY$="I" THEN GO TO 1000
940 IF M>=15 THEN LET M=M-1: GO TO 535
950 IF INKEY$="O" THEN LET Z$="P": GO SUB 1200
955 IF INKEY$="P" THEN LET X$="P": GO SUB 1300
957 IF I=M+1 THEN LET PUN=PUN+25
960 PRINT AT 12,M-1;" B";AT 13,M-1;" B"
970 GO TO 900
1000 REM *PATADA DELANTE**
1010 INK 2: PRINT AT 12,m;"HJ";AT 13,m;"I": BEEP 0.1,1
1020 PRINT AT 12,m+1;" "
1025 LET E=E-1: PRINT ; PAPER 4; INK 1;AT 19,1;"ENERGIA : ";E;" "
1027 IF D=2 AND M=MOV-2 THEN PRINT AT 12,MOV;"U": BEEP .009,40: BEEP .009,19: PRINT AT 12,MOV;" ";AT 12,MOV;" ": LET MOV=29: LET D=INT (RND*3): GO TO 960
1030 IF M=MOV-2 THEN GO SUB 8000
1040 IF D=2 AND M=MOV-2 THEN STOP
1050 GO TO 930
1100 REM *PATADA ATRAS*
1110 INK 2: PRINT AT 12,M-1;"QE";AT 13,M;"I": BEEP 0.1,1
1115 LET E=E-1: PRINT ; PAPER 4; INK 1;AT 19,1;"ENERGIA : ";E;" "
1120 PRINT AT 12,M-1;" "
1125 IF M=MOV+2 THEN GO SUB 8000
1130 GO TO 830
1200 REM SALTO
1210 PRINT AT 12,M;Z$;AT 13,M;" ": BEEP 0.5,1
1215 LET E=E-1: PRINT ; PAPER 4; INK 1;AT 19,1;"ENERGIA : ";E;" "
1220 IF I=M-1 THEN PRINT AT 13,M;D$: PRINT AT 13,M-1;" ": LET I=M+1
1230 RETURN
1300 REM ***AGACHARSE***
1310 PRINT AT 13,M;X$;AT 12,M;" ": BEEP 0.5,1
1315 LET E=E-1: PRINT ; PAPER 4; INK 1;AT 19,1;"ENERGIA : ";E;" "
1320 IF D=2 AND M=MOV-1 THEN LET MOV=M-1: PRINT AT 12,M-1;"M";AT 12,M+1;" ": LET MOV=M-1: LET PUN=PUN+25
1330 RETURN
1999 REM MOVIM.. PATADA MALO
2000 INK 0: IF MOV=2 THEN LET MOV=29
2001 IF D=2 THEN LET Y$=R$: LET J$=" "
2004 IF D<>2 THEN LET Y$="C": LET J$="D"
2005 PRINT AT 12,MOV;Y$;AT 13,MOV;J$
2007 PRINT AT 12,MOV+1;" ";AT 13,MOV+1;" "
2009 BEEP .005,19: BEEP .005,10
2013 PRINT AT 12,3;" ";AT 13,3;" "
2014 IF D=2 AND M=MOV THEN PRINT AT 12,M;"U";AT 13,M;"U": BEEP .009,40: BEEP .009,19: PRINT AT 12,M;" ";AT 13,M;" ": LET PUN=PUN+50: LET E=E-1: GO TO 2090
2015 IF E<=0 THEN GO TO 2080
2020 IF D<>2 AND MOV=M+1 THEN GO TO 2050
2030 RETURN
2050 PRINT AT 12,MOV;"QE";AT 13,MOV;" I"
2060 BEEP .09,-5
2070 PRINT AT 12,MOV-1;"  C";AT 13,MOV;" D"
2079 REM MUERTE DEL HEROE
2080 PRINT AT 13,M-1;"KL"
2085 LET VID=VID-1
2086 PRINT PAPER 4; INK 1;AT 21,1;" VIDAS : ";VID
2087 LET E=100
2090 LET X=0: LET Y=15: LET A$="22282222822224444822228442822224444"
3000 LET B$="59:<59:<59:<959799759<<:99:<9575"
3010 FOR J=1 TO LEN A$
3020 LET C=CODE A$(J)-48: LET DISCO=CODE B$(J)-48
3030 BEEP C/Y,DISCO+X: NEXT J
3040 PRINT AT 12,MOV;" ";AT 13,MOV;" "
3100 PAUSE 0: BEEP 0.05,1: PRINT AT 12,MOV+1;" ";AT 13,MOV+1;" ": LET MOV=29
3110 IF VID=0 THEN GO TO 9100
3120 GO TO 960
8000 REM CAIDA DEL NICO
8001 LET PUN=PUN+50
8002 LET D=INT (RND*3)
8004 IF D<0 OR D>3 THEN GO TO 8002
8005 PRINT AT 12,MOV;" ";AT 13,MOV;" "
8010 OVER 1
8020 INK 3: FOR U=12 TO 15: PRINT AT U,MOV;"C";AT U+1,MOV;"D"
8030 BEEP .05,U
8040 PRINT AT U,MOV;"C";AT U+1,MOV;"D"
8050 NEXT U
8060 OVER 0
8070 LET MOV=29: RETURN
8708 GO TO 800
9000 REM SERPIENTE
9010 IF D=0 OR D=2 THEN LET D$=S$
9012 IF I>MOV THEN GO TO 9020
9015 IF D=1 THEN LET D$=R$
9020 BEEP .0001,30
9040 LET I=I+1: PRINT AT 13,I;D$;AT 13,I-1;" "
9050 IF I=29 THEN LET I=1
9055 IF I=M THEN PRINT AT 13,M;"U";AT 12,M;"U": BEEP .009,40: BEEP .009,19: PAUSE 30: PRINT AT 13,M;" ";AT 12,M;" ": GO SUB 2085
9060 RETURN
9499 REM CONSOLACION
9510 GO SUB 9540
9520 INK 7: PAPER 0: PRINT AT 6,2;"DESEAS JUGAR OTRA VEZ (S/N)";AT 7,2;"============================";AT 10,2;"HAS CONSEGUIDO ";PUN;" PUNTOS";AT 14,2;"© JOSE AVILA AVILA RUIZ": LET PUN=0: LET E=100: LET VID=5
9525 IF INKEY$="S" THEN GO SUB 9600: GO TO 500
9527 IF INKEY$="N" THEN STOP
9530 GO TO 9525
9535 REM
9540 PAUSE 0: FOR a=5 TO 16: OVER 0: FOR b=1 TO 30: INK 0: BEEP 0.005,40: PRINT AT a,b;"█": NEXT b: NEXT a
9545 RETURN
9600 FOR a=16 TO 5 STEP -1: OVER 0: FOR b=1 TO 30: INK 5: BEEP 0.005,40: PRINT AT a,b;"█": NEXT b: NEXT a
9601 RETURN
9700 REM FELICITACION
9705 GO SUB 9540
9710 PRINT AT 6,3;"ERES UN BUEN SAMURAI";AT 7,3;"====================";AT 10,2;"DESEAS JUGAR OTRA VEZ (S/N)";AT 13,2;"ABAJO TIENES LA PUNTUACION";AT 15,2;"!!!!! BUENA SUERTE SOLDADO!!!!"
9730 GO TO 9524
```

MICROHOBBY

PRESENTA

El mayor éxito del año en todas las pantallas de España

NOMINADA PARA SEIS OSCARS



MEJOR PROGRAMA

Ensamblador

MEJOR GUION

Editext

MEJOR DOBLAJE

Monitor

MEJOR GRAFISTA

Screen

MEJOR MUSICA

Data Beep

MEJOR MONTAJE

Renumerador

CON EL MAYOR DESPLIEGUE DE EXTRAS DE LA HISTORIA DEL SOFT

Desensamblador, Cargador CM., Depurador, Cirujano, Tokens, Micro CAT, Conversor, Listador, Copiupi, Audio Agenda, Multi UDG, Traspa, Voz.

Proyección especial hasta el 31 de mayo para todos nuestros lectores consistente en:

«2 CINTAS que contienen 20 PROGRAMAS DE UTILIDADES valoradas en 2.500 ptas.), gratis al realizar tu suscripción»

Recorta o copia este cupón y envíalo a Hobby Press, S. A., Apartado de Correos 232. Alcobendas (Madrid).

Nombre ______ Apellidos ______
Dirección ______ C. Postal ______
Localidad ______ Provincia ______
Teléfono ______ Profesión ______

Deseo suscribirme a **MICROHOBBY SEMANAL** (50 números) al precio de 5.670 ptas. (IVA incluido). Esta suscripción me da derecho a recibir **totalmente gratis, 2 cintas,** que contienen 20 programas de utilidades, valoradas en **2.500 ptas.** (Oferta válida hasta el 31 de mayo de 1986).

Deseo recibir en mi domicilio la cinta «20 utilidades», al precio de 2.500 ptas. (suscriptores 1.900 ptas.).
Número de suscriptor ______ (si no lo recuerda escriba sólo la palabra «SI»).

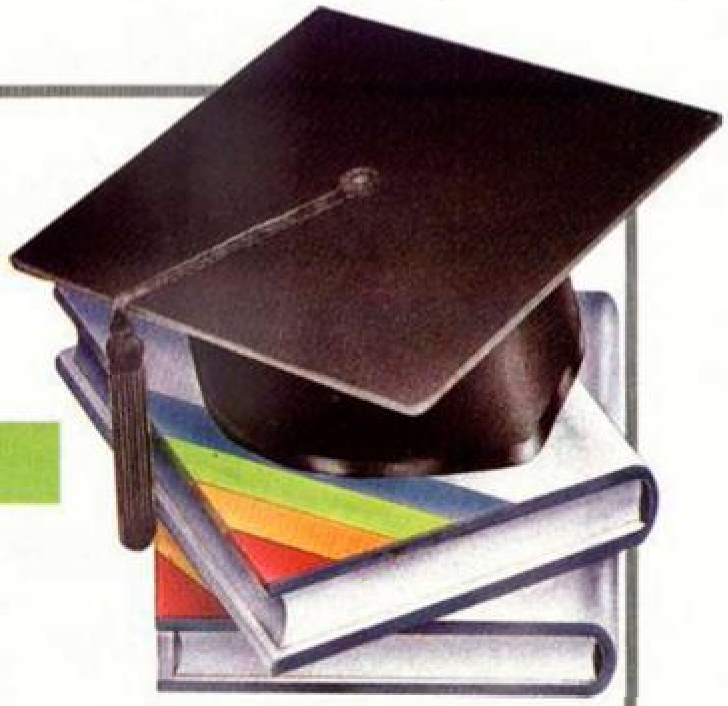
FORMA DE PAGO. MARCA CON UNA X LA OPCION QUE DESEES.

- ☐ Contra reembolso
- ☐ Mediante tarjeta VISA. Núm. de la tarjeta ______ Fecha caduc. de la tarjeta ______
- ☐ Mediante talón bancario a nombre de Hobby Press, S. A.
- ☐ Mediante giro postal n.° ______
- ☐ Mediante domiciliación bancaria

Banco ______ Sucursal y Localidad ______
N.° de cuenta ______ Fecha y firma

Profesor particular

Arturo LOBO y J. J. LEON



INTEGRALES INDEFINIDAS 2

Como ya os anunciamos en el número anterior, esta semana os ofrecemos la parte que faltaba del programa integrales indefinidas. Para cargarlo podéis hacer dos cosas; o bien teclear éste y luego dar MERGE sobre el otro, o bien cargar la primera parte en el ordenador y luego teclear estas líneas encima de las otras.

Como también explicamos, por sí sola ninguna de las dos partes funciona.

El programa ofrece un menú de cuatro opciones: integrales definidas, indefinidas, dibujo de la función y dibujo de la función integral.

Antes de dibujar, el programa tiene que haber pasado por la opción de integrales indefinidas; esta opción obtiene la $\int^{x}_{xo} f(x)\,dx$ para un intervalo dado (no en todo el eje real) y con un error dado. Este error se tiene que elegir de acuerdo con los valores previsibles de la integral, es absurdo que si la integral hasta un punto valiera 10.000, pedirle un error de 10^{-3} por que el programa podría tardar mucho tiempo y no merecer la pena la precisión ganada.

Nosotros recomendamos que se halle la integral hasta un punto mediante la opción de integrales definidas y luego, se escoja un error del 1 por 100 o por mil de ese valor, además esa operación servirá para comparar los valores obtenidos en las dos opciones.

El programa deja definida la función FNi (x) que es la función integral. Siempre vuelve al menú, pero si queréis explorar la función i (x) basta con dar BREAK en el menú y GO TO 3830 cuando queráis volver.

La variable tiempo que aparece en la línea 4000 es una medida de lo que tarda el programa; aumentando ésta podéis aumentar la precisión y, en algunos casos, incluso disminuir el tiempo de ejecución cuando el intervalo de integración es grande. Probar con funciones cuya primitiva conozcáis y comprobar la precisión del programa y los tiempos de ejecución, que tenemos que decir que pueden ser elevados, de varios minutos en general.

Podéis, además, ver dibujada tanto la función como su integral en pantalla con las opciones 3 y 4, aunque para ello tenéis que dar MERGE de este programa sobre el de «Representación Gráfica» que os ofrecimos en números pasados; comprobar que las líneas 7000 de este programa son las del listado.

Cuando escribáis la función a través del Spectrum, tenéis que tener en cuenta que hay que hacerlo con las funciones del teclado y que la operación potencia del Spectrum no permite elevar un número negativo a otro número, por lo que puede dar error si las funciones introducidas toman valores negativos elevados a algo.

TODAS LAS LETRAS MAYUSCULAS SUBRAYADAS DEBERAN TECLEARSE EN MODO GRAFICO

```
8000 DEF FN l(x)=x*((x>64)*(x<91)+(x>96)*(x<123)+(x>47)*(x<58))
8001 DEF FN p(x)=3*((x=43)+(x=45))+2*((x=42)+(x=47))+(x=94)
8005 LET d$=e$: GO SUB 8480: IF pvar=0 THEN LET e$="0": RETURN
8007 LET a$=e$: LET paren=0: GO SUB 8040: LET e$=a$
8010 LET b$="": FOR o=1 TO LEN e$
8012 IF CODE e$(o)<>0 THEN LET b$=b$+e$(o): NEXT o: GO TO 8024
8014 LET np=0: FOR i=o+1 TO LEN e$: IF CODE e$(i)=40 THEN LET np=np+1
8016 IF CODE e$(i)=41 THEN LET np=np-1: IF np=0 THEN GO TO 8020
8018 NEXT i: PRINT FLASH 1;"numero impar de parentesis": STOP
8020 LET nuevao=i+1: LET a$=e$(o+2 TO i-1): LET paren=0: GO SUB 8040
8021 IF a$="1" THEN IF b$(LEN b$)="*" THEN LET b$=b$(1 TO LEN b$-1): GO TO 8023
8022 LET b$=b$+a$
8023 LET o=nuevao: IF o<=LEN e$ THEN GO TO 8012
8024 IF b$=e$ THEN RETURN
8026 LET e$=b$: GO TO 8010
8039 LET paren=1
8040 DIM a(25,2): DIM b(25): LET ax=1: LET np=0: LET a(1,2)=0: FOR i=1 TO LEN A$: IF CODE A$(i)=40 THEN LET np=np+1: IF np=1 THEN LET ax=ax+1: LET a(ax,1)=i: NEXT i
8045 IF CODE A$(i)=41 THEN LET np=np-1: IF np=0 THEN LET a(ax,2)=i: NEXT i
8047 NEXT i: IF np<>0 THEN PRINT "numero impar de parentesis !!!": STOP
8050 LET a(ax+1,1)=LEN A$+1: IF a(2,1)=1 THEN IF a(2,2)=LEN A$ THEN LET A$=A$(2 TO (LEN A$-1)): GO TO 8039
8055 LET no=0: LET niv=0: LET b(1)=0: FOR j=1 TO ax: FOR i=a(j,2)+1 TO a(j+1,1)-1: LET orden=FN p(CODE a$(i)): IF orden>niv THEN LET niv=orden: LET no=2: LET b(2)=i: GO TO 8062
8060 IF orden=niv THEN IF niv>0 THEN LET no=no+1: LET b(no)=i
8062 NEXT i: NEXT j: LET b(no+1)=LEN A$+1
8070 GO SUB (8100+100*niv): RETURN
8100 IF a$=i$ THEN LET a$="1": RETURN
8102 LET solo=0: IF a$(2 TO LEN a$)=i$ THEN LET solo=1
8105 GO TO (7930+CODE a$(1))
8108 LET a$="COS "+a$(2 TO LEN a$)+"*"+CHR$ 0+"("+a$(2 TO LEN a$)+")": GO TO 8130
8109 LET a$="(-SIN "+a$(2 TO LEN a$)+")"+"*"+CHR$ 0+"("+a$(2 TO LEN a$)+")": GO TO 8130
8110 LET a$="1/ABS COS "+a$(2 TO LEN a$)+"↑2"+"*"+CHR$ 0+"("+a$(2 TO LEN a$)+")": GO TO 8130
8111 LET a$="1/SQR (1-ABS "+a$(2 TO LEN a$)+"↑2)"+"*"+CHR$ 0+"("+a$(2 TO LEN a$)+")": GO TO 8130
8112 LET a$="(-1)/SQR (1-ABS "+a$(2 TO LEN a$)+"↑2)"+"*"+CHR$ 0+"("+a$(2 TO LEN a$)+")": GO TO 8130
8113 LET a$="1/(1+ABS "+a$(2 TO LEN a$)+"↑2)"+"*"+CHR$ 0+"("+a$(2 TO LEN a$)+")": GO TO 8130
8114 LET a$="1/"+a$(2 TO LEN a$)+"*"+CHR$ 0+"("+a$(2 TO LEN a$)+")": GO TO 8130
8115 LET a$=a$+"*"+CHR$ 0+"("+a$(2 TO LEN a$)+")": GO TO 8130
8117 LET a$="1/2/"+a$+"*"+CHR$ 0+"("+a$(2 TO LEN a$)+")": GO TO 8130
8119 LET a$=CHR$ 0+"("+a$(2 TO LEN a$)+")": GO TO 8130
8130 IF solo=1 THEN LET a$=a$(1 TO (LEN a$-LEN i$-4))
8131 RETURN
8200 LET c$=a$(b(2)+1 TO b(3)-1): FOR i=3 TO no: LET c$=c$+"+"+a$(b(i)+1 TO b(i+1)-1): NEXT i: IF no>=3 THEN LET c$="("+c$+")"
8210 LET d$=a$(1 TO b(2)-1): GO SUB 8480: LET pvar1=pvar: LET d$=c$: GO SUB 8480: LET pvar2=pvar: IF pvar1*pvar2<>0 THEN LET a$=a$(1 TO b(2)-1)+"↑"+c$+"*LN "+a$(1 TO b(2)-1)+"*"+CHR$ 0+"("+c$+"+"+c$+"/"+a$(1 TO b(2)-1)+"*"+CHR$ 0+"("+a$(1 TO b(2)-1)+")": RETURN
8220 IF pvar2<>0 THEN LET a$=a$(1 TO b(2)-1)+"↑"+c$+"*LN "+a$(1 TO b(2)-1)+"*"+CHR$ 0+"("+c$+")": RETURN
8222 DEF FN s(x)=FN l(x)+((x<48)+(x>58))
8225 LET tra=0: FOR i=1 TO LEN c$: LET tra=tra+FN s(CODE c$(i)): NEXT i: IF tra=0 THEN LET c$=STR$ (VAL c$): IF VAL c$=0 THEN LET a$="0": RETURN
8227 IF c$="1" THEN LET a$=CHR$ 0+"("+a$(1 TO b(2)-1)+")": RETURN
8230 LET a$=c$+"*"+a$(1 TO b(2)-1)+"↑("+c$+"-1)"+"*"+CHR$ 0+"("+a$(1 TO b(2)-1)+")": RETURN
8300 LET c$=""
8310 FOR i=1 TO no: LET d$=a$(b(i)+1 TO b(i+1)-1): GO SUB 8480: IF pvar=0 THEN NEXT i: GO TO 8350
8320 LET d$="": FOR j=1 TO no: IF j<>i THEN LET d$=d$+a$(b(j)+(j=1) TO b(j+1)-1): NEXT j: GO TO 8340
8325 IF a$(b(i)+(i=1))="/" THEN LET s9=45: LET d$=d$+"/ABS ("+a$(b(i)+1 TO b(i+1)-1)+")↑2*"+CHR$ 0+"("+a$(b(i)+1 TO b(i+1)-1)+")": NEXT j: GO TO 8340
8330 LET s9=43: LET d$=d$+"*"+CHR$ 0+"("+a$(b(j)+1 TO b(j+1)-1)+")": NEXT j
8340 LET c$=c$+CHR$ s9+d$: NEXT i
8350 IF (c$(1 TO 2)="+*")+(c$(1 TO 2)="-*")=1 THEN LET c$=c$(3 TO LEN c$)
8352 IF c$(1)="+" THEN LET c$=c$(2 TO LEN c$)
8355 IF paren=1 THEN LET a$="("+c$+")": RETURN
8358 LET a$=c$: RETURN
8400 LET d$=a$(b(1)+1 TO b(2)-1): GO SUB 8480: IF pvar<>0 THEN LET c$=CHR$ 0+"("+d$+")": GO TO 8410
8405 LET c$=""
8410 FOR i=2 TO no: LET d$=a$(b(i)+1 TO b(i+1)-1): GO SUB 8480: IF pvar<>0 THEN LET c$=c$+a$(b(i))+CHR$ 0+"("+d$+")"
8415 NEXT i: IF paren=1 THEN LET a$="("+c$+")": RETURN
8420 LET a$=c$: RETURN
8480 IF d$="" THEN LET pvar=0: RETURN
8490 LET pvar=0: LET fin=LEN d$-LEN i$+1: FOR p=1 TO fin: IF d$(p TO (p-1+LEN i$))=i$ THEN IF ((FN l(CODE d$(p-1+(p=1)))=0)+(p=1))*((FN l(CODE d$(p+LEN i$-(p=fin)))=0)+(p=fin))=1 THEN LET pvar=p: RETURN
8492 NEXT p: RETURN
```



WINTER SPORTS • Deportivo • Electric Dreams

DIVERSION EN LA NIEVE

Un nuevo juego de simulación deportiva inscribe su nombre en la ya amplia lista del software, Juegos de Invierno.

Mucho de razón tiene el refrán que dice aquello de «no es oro todo lo que reluce», y este Winter Sports es una buena prueba de ello. Este programa «reluce» con bastante fulgor debido al enorme atractivo que representa el poder disfrutar de ocho pruebas diferentes en un mismo juego y además, tratándose de deportes tan interesantes y espectaculares.

Sin embargo, observandolo más de cerca, comprobamos que se trata de un oro de no muchos kilates. La verdad es que no existe una línea uniforme de adicción entre todas las pruebas, ya que la gracia y diversión de las mismas varía notable mente entre unas y otras, pudiéndo nos encontrar desde deportes bastante adictivos y bien realizados, hasta otros relativamente pobres y demasiado lentos, por lo que se pierde gran parte de la emoción y sensación de realismo.

Pero pasemos a comentar uno por uno los eventos que componen la totalidad del programa y que, algo muy de agradecer, pueden ser cargados independientemente, lo que nos permite acceder directamente a la prueba sin necesidad de pasar por todas las anteriores.

1. Downhill (Descenso). Esta prueba es similar a las dos siguientes y son las que menos calidad ofrecen debido a la lentitud con la que se produce la bajada por la montaña. Lo que tendremos que hacer es ir pasando entre las numerosas puertas que se encuentran en la nieve y llegar a la meta lo antes posible.

2. Slalom. Consiste en sortear de izquierda a derecha, alternativamente, los palos que se encuentran clavados en la nieve intentando cometer el menor número posible de fallos. El tiempo también contará en la puntuación final.

3. Giant Slalom (Slalom Gigante). Esta prueba es la unión de las dos anteriores, por lo que la distancia a recorrer es bastante considerable y resultará más complicado completar todo el recorrido sin cometer ningún error.

Como podréis comprobar en las fotografías, en estas tres pruebas la pantalla está dividida en dos secciones principales, en las que se nos muestra la visión que nosotros tenemos en los descensos y una vista aérea del recorrido que estamos realizando. En la parte superior aparece un marcador que indica el número de puertas pasadas o falladas y el tiempo que llevamos.

4. Ice Hockey (Hockey sobre hielo). Esta es una de las pruebas más interesantes y divertidas, puesto que tenemos la oportunidad de vivir toda la emoción de un auténtico encuentro de hockey. La técnica a seguir con este deporte es muy similar a la del World Series Basketball, con la diferencia de que en lugar de ir botando una pelota, deberemos deslizarnos rápidamente tras un pequeño disco.

5. Ski Jumping (Salto). Esta consiste en bajar a toda velocidad por un largo tobogán y una vez llegado al final del mismo, tomar impulso e intentar llegar lo más lejos posible. Este deporte seguramente os será muy familiar a todos, no porque lo hayáis practicado, sino porque cada primero de año, tenemos la posibilidad de disfrutar de la emoción de esta prueba a través del televisor.

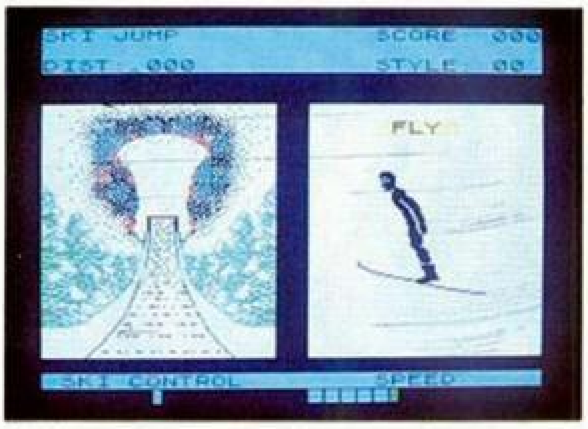
6. Speed Skatimg (Patinaje de velocidad). Deberemos tratar de alcanzar la máxima velocidad posible simplemente empujándonos con nuestros patines y llegar a la meta antes que nuestro contrincante.

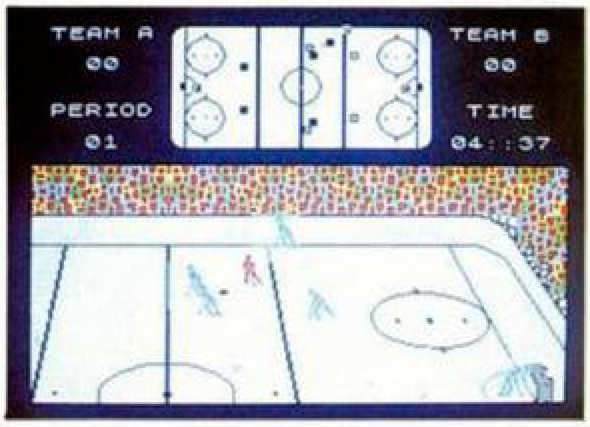
7. Bobsled. Consiste en bajar por un largo pasillo metidos en nuestro aerodinámico trineo intentando no salirnos del camino para no chocar contra alguno de los













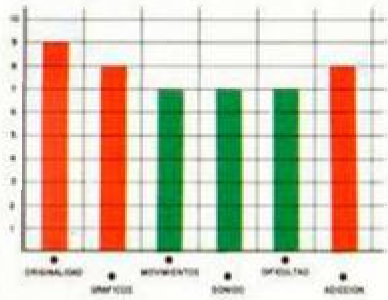


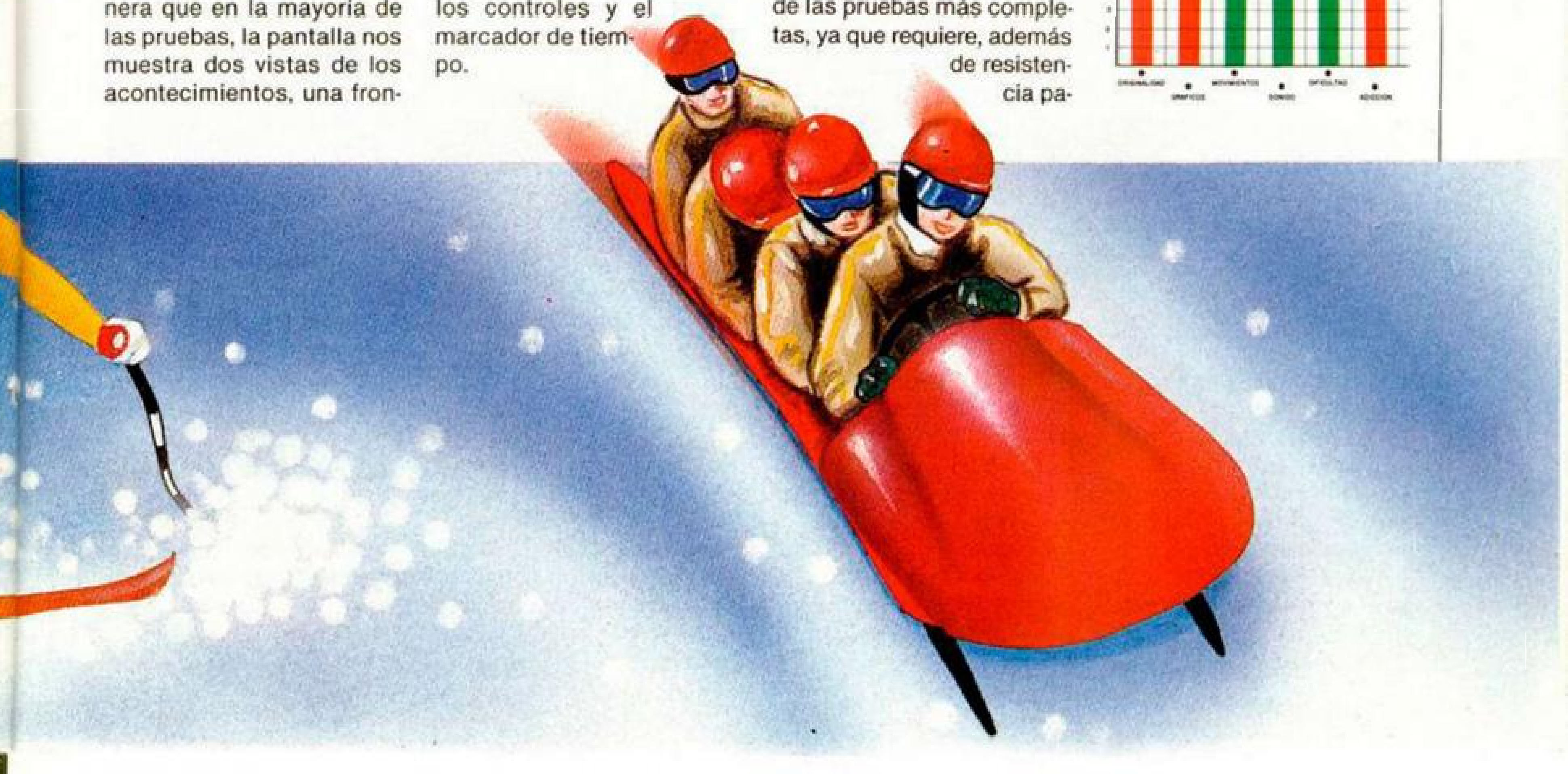
laterales. De la misma manera que en la mayoría de las pruebas, la pantalla nos muestra dos vistas de los acontecimientos, una frontal y otra lateral, además de los controles y el marcador de tiempo.

8. Biathlon. Esta es una de las pruebas más completas, ya que requiere, además de resistencia para poder deslizarnos por terrenos bastante escarpados, la suficiente puntería para conseguir una buena puntación en las pruebas de tiro que deberemos afrontar durante el desarrollo de la misma. Tanto el tiempo empleado como las dianas realizadas, influirán notablemente en el resultado final.

Como podéis comprobar, mayor variedad no se le puede pedir a un juego, pero como antes decíamos, debido a esta misma variedad, se ha perdido un poco de calidad, y a pesar de que la mayoría de las pruebas resultan más que aceptables, como el hockey, Biathlon, Skating, etc..., hay alguna que otra, como los Slalom en general, que han quedado bastante flojos en el aspecto de la diversión.

Sin embargo, a pesar de estos pequeños fallos, Winter Sports es un programa que tanto en sus aspectos gráficos como en su originalidad, vistosidad y emoción en gran parte del mismo, reúne las características necesarias para ser una auténtica estrella.





TOY BIZARRE • Arcade • Activision

LA REBELION DE LOS MUÑECOS



Este Toy Bizarre llega con bastante retraso a nuestro país, pues fue realizado hace ya algunos años por Activision. El juego consiste en un movido arcade en el cual Merton, el guardián nocturno de una fábrica de juguetes, debe tratar de parar a los muñecos que se han rebelado y que andan desperdigados por todas partes.

El juego comienza con cuatro vidas, durante el transcurso de las cuales debemos tratar de cerrar cuatro de seis válvulas que aparecen en cada pantalla que son las que van llenando unos globos que, si no los destruimos antes de que exploten por sí solos, se convertirán en los liberadores de estos rebeldes y destructivos juguetes.

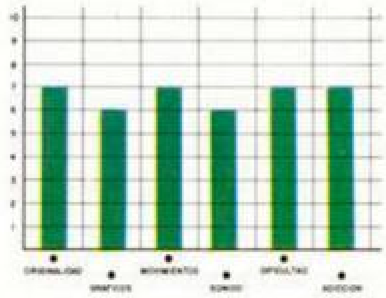
Además, tenemos que contar con que el travieso Hefty Hilda, nos perseguirá incansablemente y que intentará cerrar las válvulas que hayamos conseguido abrir, y que será nuestro mayor enemigo durante todo el transcurso del juego.

Toy Bizarre es un juego bastante rápido en el que tenemos que estar continuamente pendientes de los globos, de las válvulas, de Hefty y de unas plataformas que también nos sirven de alguna ayuda, por lo que el ritmo al que Merton tiene que moverse por las diferentes pantallas es muy rápido y no podemos perder ni un instante en nuestra frenética carrera de saltos, subidas, bajadas, y persecuciones, para conseguir que el orden reine por fin en esta enloquecedora fábrica.



Sus gráficos no son excesivamente buenos, pero no resultan desagradables y no desentonan demasiado con el nivel general del programa. Por lo que, en resumen, Toy Bizarre es un juego que puede agradar a los adictos a los arcades y que, además, también crea bastante adicción.

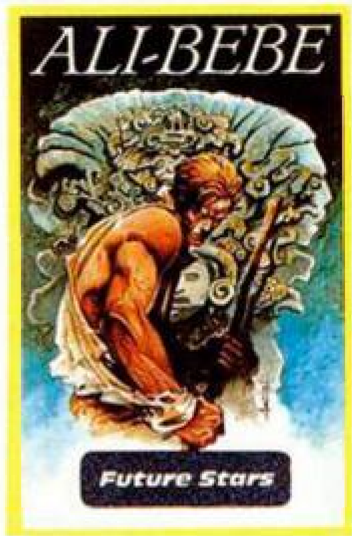
Por último, podríamos destacar el dominio que tenemos sobre los movimientos del personaje, el cual, además de poder dar grandes saltos puede dirigir sus movimientos en el aire.



ALI BEBE • Videoaventura • Future Star

EL LABERINTO MALDITO

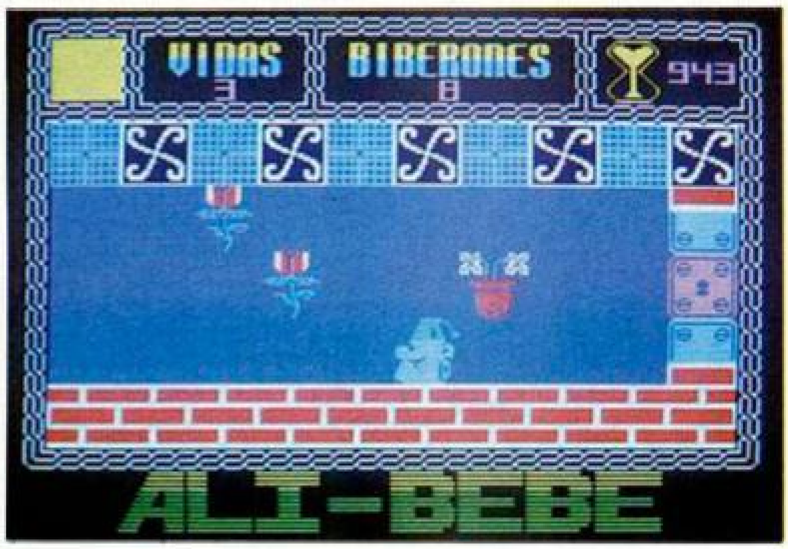
Este Alí-Bebé es uno de los tres primeros títulos que han hecho su aparición bajo el nuevo sello de Dinamic, llamado Future Star, el cual tiene como principal objetivo servir de medio a todos aquellos programadores noveles que desean dar a conocer sus primeras creaciones. Se trata de una videoaventura con una concepción muy similar, tanto en sus aspectos gráficos como en su desarrollo, a anteriores programas de Dinamic como Saimazoon o, muy especialmente a Babaliba.



En Alí-Bebé el protagonista es un pequeño huerfanito árabe, cuyo malvado padre decidió encerrarle en un complicado laberinto plagado de peligros y alimañas. Para encontrar el camino de salida, este morito tendrá que dar con tres objetos mágicos: una lámpara mágica, una espada y un chupete. Una vez que haya conseguido hacerse con todo, deberá llevarlos a una habitación y colocarlos adecuadamente. Los objetos se encuentran, a su vez, encerrados en sus correspondientes salas cerradas, por lo que Alí tendrá que encontrar previamente las llaves respectivas para abrir sendas puertas.

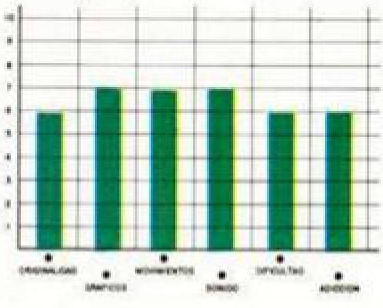


Para que el tortuoso ca-



mino en busca de la libertad le resulte un poco más sencillo, Alí-Bebé puede contar con algunas notables ayudas. Por ejemplo, cuenta con una buena reserva de biberones con los que podrá defenderse de sus atacantes y también existen repartidos por el laberinto algunos transportadores que le llevan a lugares cerrados, aunque esto no ocurrirá siempre que lo desee.

Un juego entretenido y bonito en sus aspectos gráficos.



THE FOURTH PROTOCOL • Estrategia
Century Communications

DESVELA EL PLAN AURORA



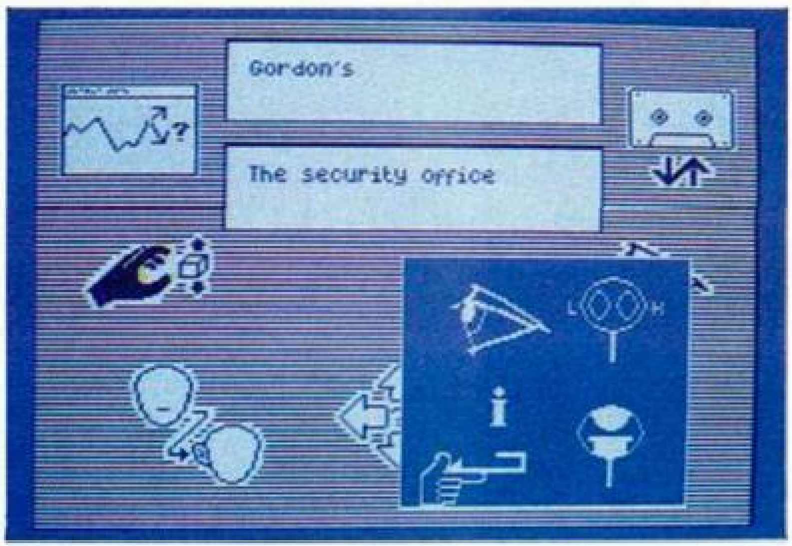
Este programa llegó a nuestras manos hace ya algún tiempo, pero debido a lo significativo de estas fechas, en las que el controvertido tema de la OTAN acapara la atención de todos, hemos decidido sacar de nuestro archivo este The Fourth Protocol, ya que su argumento está íntimamente relacionado con este asunto de la Alianza Atlántica.

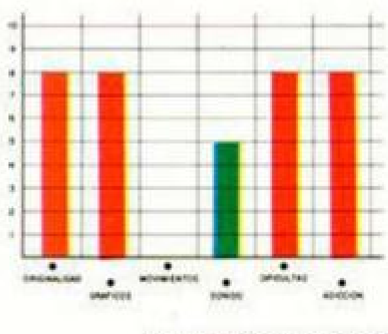
Se trata de un juego de estrategia basado en una obra de Frederich Forsyth, un conocido escritor de novelas de acción, y en el cual representamos el papel de un hábil agente de la CIA llamado John Preston.

Nuestra misión consistirá en recomponer las desbaratadas piezas del rompecabezas que los agentes enemigos han establecido para encubrir el sabotaje que, tiempo atrás, planearon en las afueras de Moscú y que dieron en llamar Plan Aurora. Para ello, tendremos la posibilidad de acceder, gracias a un sencillo y cómodo sistema de iconos, a una innumerable cantidad de acciones diferentes tales como obtener información de la central de comunicaciones, utilizar los servicios de inteligencia, realizar y recibir llamadas telefónicas, investigar archivos, contratar espías... y una interminable lista de posibilidades que hacen de The Fourth Protocol un juego verdaderamente interesante para los amantes de la estrategia y de los temas de espionaje.

Evidentemente, debido a la gran complejidad del programa, tardaremos algún tiempo en llegar a conocer en profundidad las particularidades del juego, y posiblemente las primeras veces que nos enfrentemos a tan complicada misión sacaremos muy pocas cosas en claro, pero a medida que nos vayamos compenetrando con nuestro agente John Preston, iremos confirmando la idea de que El Cuarto Protocolo nos va a asegurar muchas horas de intriga, emoción y diversión que es de lo que se trata.







CODIGO DE MAQUINA:

1 1 0 0 1 0 1 1	CBh
0 0 0 0 1 1 1 0	0Eh

INDICADORES DE CONDICION QUE AFECTA:

S; pone 1 - si el resultado es negativo;
pone 0 - en cualquier otro caso
Z; pone 1 - si el resultado es cero;
pone 0 - en cualquier otro caso
H; pone 0 - siempre
N; pone 0 - siempre
C; pone el valor que tenía el bit 0 del octeto antes de la ejecución
P/V; pone 1 - si la paridad es par;
pone 0 - en cualquier otro caso

CICLOS DE MEMORIA:

4

CICLOS DE RELOJ:

15

EJEMPLO:

RRC (HL)

Contenido del par de registros «HL»

(H):	0 1 1 1 0 0 0 0	70h
(L):	0 0 1 0 0 1 1 1	27h

Contenido del octeto de memoria 7027h

7027h: 0 0 0 0 0 0 0 1 — 01h

Instrucción

RRC (HL):	1 1 0 0 1 0 1 0	CBh
	0 0 0 0 1 1 1 0	0Eh

Contenido del octeto de memoria 7027h después de la ejecución

7027h: 1 0 0 0 0 0 0 0 — 80h

El contenido del par de registros «HL» no ha variado

Indicadores de condición después de la ejecución

S	Z		H		P/V	N	C
1	0	x	0	x	0	0	1

RRC (IX + d)

OBJETO:

Rota a la derecha, un bit, el octeto de memoria direccionado por el contenido del registro índice «IX» más el entero de desplazamiento «d», el cual puede adquirir los valores desde —128 a + 127. El valor del bit 0 saliente se copia en el indicador de acarreo «C» y en el bit 7 entrante. Ver Figura 9-3.

CODIGO DE MAQUINA:

1 1 0 1 1 1 0 1	DDh
1 1 0 0 1 0 1 1	CBh
← d →	
0 0 0 0 1 1 1 0	0Eh

INDICADORES DE CONDICION QUE AFECTA:

S; pone 1 - si el resultado es negativo;
pone 0 - en cualquier otro caso
Z; pone 1 - si el resultado es cero;
pone 0 - en cualquier otro caso
H; pone 0 - siempre
N; pone 0 - siempre
C; pone el valor que tenía el bit 0 del octeto antes de la ejecución
P/V; pone 1 - si la paridad es par;
pone 0 - en cualquier otro caso

CICLOS DE MEMORIA:

6

CICLOS DE RELOJ:

23

EJEMPLO:

RRC (IX + 43)

Contenido del registro índice «IX»

(IX):	1 0 0 0 0 1 0 1	85h
	0 0 1 0 0 1 1 1	27h

Contenido del octeto de memoria 856Ah (IX + 43)

856Ah: 0 0 1 0 1 0 1 0 — 2Ah

Instrucción

RRC (IX + 43):	1 1 0 1 1 1 0 1	DD
	1 1 0 0 1 0 1 0	CBh
	0 0 1 0 1 0 1 1	2Bh
	0 0 0 0 1 1 1 0	0Eh

Contenido del octeto de memoria 856Ah después de la ejecución

856Ah: 0 0 0 1 0 1 0 1 — 15h

Indicadores de condición después de la ejecución

S	Z		H		P/V	N	C
0	0	x	0	x	0	0	0

RRC (IY + d)

OBJETO:

Rota a la derecha, un bit, el octeto de memoria direc-

RRA

OBJETO:

Rota a la derecha el contenido del registro acumulador un bit. El contenido del bit 0 saliente se copia en el indicador de acarreo y el indicador de acarreo anterior se copia en el bit 7 entrante (Rotate Right Acumulator). Ver Figura 9-4.

CODIGO DE MAQUINA:

0 0 0 1 1 1 1 1 — 1Fh

INDICADORES DE CONDICION QUE AFECTA:

H; pone 0 - siempre
N; pone 0 - siempre
C; pone el valor que tenía el bit 0 del registro A antes de la ejecución

CICLOS DE MEMORIA:

1

CICLOS DE RELOJ:

4

EJEMPLO:

RRA

Contenido del registro acumulador

(A): 1 1 1 1 0 0 0 0 — F0h

Indicador de acarreo C = 0

Instrucción

RRA: 0 0 0 1 1 1 1 1 — 1Fh

Contenido del registro «A» después de la ejecución

(A): 0 1 1 1 1 0 0 0 — 78h

Indicadores de condición después de la ejecución

S	Z		H		P/V	N	C
x	x	x	0	x	x	0	0

RLC r

OBJETO:

Rota a la izquierda el contenido del registro representado por «r» un bit. El contenido del bit 7 saliente se copia en el bit 0 entrante y en el indicador de acarreo «C» (Rotate Left with Carry «r»). El código de representación de «r» es el señalado más abajo. Ver Figura 9-1.

Registro	Código
B	0 0 0
C	0 0 1
D	0 1 0
E	0 1 1
H	1 0 0
L	1 0 1
A	1 1 1

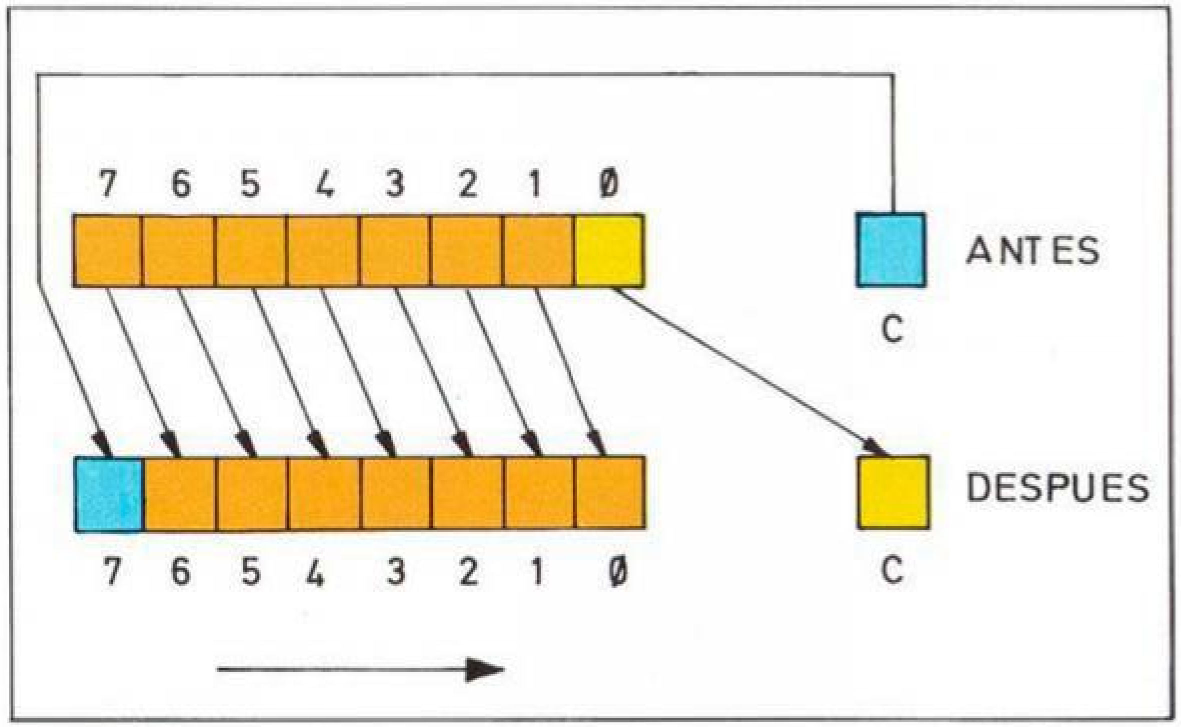


Fig. 9-4. Instrucciones RPA y RR.

CODIGO DE MAQUINA:

1 1 0 0 1 0 1 1	CBh
0 0 0 0 0 ←r→	

INDICADORES DE CONDICION QUE AFECTA:

S; pone 1 - si el resultado es negativo;
pone 0 - en cualquier otro caso
Z; pone 1 - si el resultado es cero;
pone 0 - en cualquier otro caso
H; pone 0 - siempre
N; pone 0 - siempre
C; pone el valor que tenía el bit 7 del registro r antes de la ejecución
P/V; pone 1 - si la paridad es par;
pone 0 - en cualquier otro caso

CICLOS DE MEMORIA:

2

CICLOS DE RELOJ:

8

EJEMPLO:

RLC D

Contenido del registro «D»

(D): 1 0 1 0 1 0 1 0 AAh

Instrucción

RLC D:	1 1 0 0 1 0 1 0	CBh
	0 0 0 0 0 0 1 0	02h

Contenido del registro «D» después de la ejecución

(D): 0 1 0 1 0 1 0 1 55h

Indicadores de condición después de la ejecución

S	Z		H		P/V	N	C
0	0	x	0	x	1	0	1

RLC (HL)

OBJETO:

Rota a la izquierda, un bit, el octeto de memoria direccionado por el contenido del par de registros «HL». El contenido del bit 7 saliente se copia en el bit 0 entrante y en el indicador de acarreo «C» (Rotate Left with Carry (HL)). Ver Figura 9-1.

CODIGO DE MAQUINA:

1 1 0 0 1 0 1 1	CBh
0 0 0 0 0 1 1 0	06h

INDICADORES DE CONDICION QUE AFECTA:

S; pone 1 - si el resultado es negativo;
pone 0 - en cualquier otro caso
Z; pone 1 - si el resultado es cero;
pone 0 - en cualquier otro caso
H; pone 0 - siempre
N; pone 0 - siempre
C; pone el valor que tenía el bit 7 del octeto antes de la ejecución
P/V; pone 1 - si la paridad es par;
pone 0 - en cualquier otro caso

CICLOS DE MEMORIA:

4

CICLOS DE RELOJ:

15

EJEMPLO:

RLC (HL)

Contenido del par de registros «HL»

(H):	0 1 1 1 0 0 1 1	73h
(L):	0 1 0 0 0 0 1 0	42h

Contenido del octeto de memoria 7342h

(7342h): 0 1 0 1 0 1 0 1 55h

Instrucción

RLC (HL):	1 1 0 0 1 0 1 0	CBh
	0 0 0 0 0 1 1 0	06h

Contenido del octeto de memoria 7342h después de la ejecución

(7342h): 1 0 1 0 1 0 1 0 AAh

El contenido del par de registros «HL» no ha variado

Indicadores de condición después de la ejecución

S	Z		H		P/V	N	C
1	0	x	0	x	1	0	0

RLC (IX+d)

OBJETO:

Rota a la izquierda, un bit, el octeto de memoria direccionado por el contenido del registro índice «IX» más el entero de desplazamiento «d», el cual puede adquirir los valores desde —128 a +127. El valor del bit 7 saliente se copia en el bit 0 entrante y en el indicador de acarreo «C». Ver Figura 9-1.

CODIGO DE MAQUINA:

1 1 0 1 1 1 0 1	DDh
1 1 0 0 1 0 1 1	CBh
←—d—→	
0 0 0 0 0 1 1 0	06h

pone 0 - en cualquier otro caso
Z; pone 1 - si el resultado es cero;
pone 0 - en cualquier otro caso
H; pone 0 - siempre
N; pone 0 - siempre
C; pone el valor que tenía el bit 7 del octeto antes de la ejecución
P/V; pone 1 - si la paridad es par;
pone 0 - en cualquier otro caso

CICLOS DE MEMORIA:

6

CICLOS DE RELOJ:

23

EJEMPLO:

RL (IY+0)

Contenido del registro índice «IY»

(IY):	1 1 1 1 0 1 0 1	F5h
	1 1 0 0 0 0 1 1	C3h

Contenido del octeto de memoria F5C3h

(F5C3h): 1 1 0 0 0 1 0 1 C5h

Indicador de acarreo C = 1

Instrucción

RL (IY+0):	1 1 0 1 1 1 0 1	DD
	1 1 0 0 1 0 1 0	CBh
	0 0 0 0 0 0 0 0	00h
	0 0 0 1 0 1 1 0	16h

Contenido del octeto de memoria F5C3h después de la ejecución

(F5C3H): 1 0 0 0 1 0 1 1 8Bh

Indicadores de condición después de la ejecución

S	Z		H		P/V	N	C
1	0	x	0	x	1	0	1

RRC r

OBJETO:

Rota a la derecha, un bit, el contenido del registro representado por «r». El contenido del bit 0 saliente se copia en el indicador de acarreo «C» y en el bit 7 entrante. Ver Figura 9-3. El código de representación de «r» es el señalado más abajo.

Registro	Código
B	000
C	001
D	010
E	011
H	100
L	101
A	111

CODIGO DE MAQUINA:

1 1 0 0 1 0 1 1	CBh
0 0 0 0 1 ←r→	

INDICADORES DE CONDICION QUE AFECTA:

S; pone 1 - si el resultado es negativo;
pone 0 - en cualquier otro caso
Z; pone 1 - si el resultado es cero;
pone 0 - en cualquier otro caso
H; pone 0 - siempre
N; pone 0 - siempre
C; pone el valor que tenía el bit 0 del registro r antes de la ejecución
P/V; pone 1 - si la paridad es par;
pone 0 - en cualquier otro caso

CICLOS DE MEMORIA:

2

CICLOS DE RELOJ:

8

EJEMPLO:

RRC L

Contenido del registro «L»

(L): 1 0 0 1 0 1 1 0 96h

Instrucción

RRC L:	1 1 0 0 1 0 1 0	CBh
	0 0 0 0 1 1 0 1	0Dh

Contenido del registro «L» después de la ejecución

(L): 0 1 0 0 1 0 1 1 4BH

Indicadores de condición después de la ejecución

S	Z		H		P/V	N	C
0	0	x	0	x	1	0	0

RRC (HL)

OBJETO:

Rota a la derecha, un bit, el octeto de memoria direccionado por el contenido del par de registros «HL». El contenido del bit 0 saliente se copia en el indicador de acarreo «C» y en el bit 7 entrante. Ver Figura 9-3.

(H):	0 1 1 0 0 0 1 1	63h
(L):	0 0 1 1 0 1 1 0	36h

Contenido del octeto de memoria 6336h

6336h:	0 0 0 0 0 0 0 0	0 0h

Indicador de acarreo C = 1

Instrucción

RL (HL)	1 1 0 0 1 0 1 0	CBh
	0 0 0 1 0 1 1 0	16h

Contenido del octeto de memoria 6336h después de la ejecución

6336h:	0 0 0 0 0 0 0 1	01h

El contenido del par de registros «HL» no ha variado

Indicadores de condición después de la ejecución

S	Z		H		P/V	N	C
0	0	x	0	x	0	0	0

RL (IX + d)

OBJETO:

Rota a la izquierda, un bit, el octeto de memoria direccionado por el contenido del registro índice «IX» más el entero de desplazamiento «d», el cual puede adquirir los valores desde —128 a +127. El valor del bit 7 saliente se copia en el indicador de acarreo «C» y el indicador de acarreo anterior se copia en el bit 0 entrante. Ver Figura 9-2.

CODIGO DE MAQUINA:

1 1 0 1 1 1 0 1	DDH
1 1 0 0 1 0 1 1	CBh
←— d —→	
0 0 0 1 0 1 1 0	16h

INDICADORES DE CONDICION QUE AFECTA:

S; pone 1 - si el resultado es negativo;
pone 0 - en cualquier otro caso
Z; pone 1 - si el resultado es cero;
pone 0 - en cualquier otro caso
H; pone 0 - siempre
N; pone 0 - siempre
C; pone el valor que tenía el bit 7 del octeto antes de la ejecución
P/V; pone 1 - si la paridad es par;
pone 0 - en cualquier otro caso

CICLOS DE MEMORIA:

6

CICLOS DE RELOJ:

23

EJEMPLO:

RL (IX+40)

Contenido del registro índice «IX»

(IX):	1 0 0 0 0 0 1 0	82h
	0 0 0 0 0 0 0 0	0 0h

Contenido del octeto de memoria 8228h

8228h:	1 1 1 1 1 1 1 1	FFh

Indicador de acarreo C = 0

Instrucción

RL (IX+40):	1 1 0 1 1 1 0 1	DD
	1 1 0 0 1 0 1 0	CBh
	0 0 1 0 1 0 0 0	28h
	0 0 0 1 0 1 1 0	16h

Contenido del octeto de memoria 8228h después de la ejecución

8228h:	1 1 1 1 1 1 1 0	FEh

Indicadores de condición después de la ejecución

S	Z		H		P/V	N	C
1	0	x	0	x	0	0	1

RL (IY + d)

OBJETO:

Rota a la izquierda, un bit, el octeto de memoria direccionado por el contenido del registro índice «IY» más el entero de desplazamiento «d», el cual puede adquirir los valores desde —128 a +127. El valor del bit 7 saliente se copia en el indicador de acarreo «C» y el acarreo anterior se copia en el bit 0 entrante. Ver Figura 9-2.

CODIGO DE MAQUINA:

1 1 1 1 1 1 0 1	FDH
1 1 0 0 1 0 1 1	CBh
←— d —→	
0 0 0 1 0 1 1 0	16h

INDICADORES DE CONDICION QUE AFECTA:

S; pone 1 - si el resultado es negativo;

INDICADORES DE CONDICION QUE AFECTA:

S; pone 1 - si el resultado es negativo
pone 0 - en cualquier otro caso
Z; pone 1 - si el resultado es cero
pone 0 - en cualquier otro caso
H; pone 0 - siempre
N; pone 0 - siempre
C; pone el valor que tenía el bit 7 del octeto antes de la ejecución
P/V; pone 1 - si la paridad es par
pone 0 - en cualquier otro caso

CICLOS DE MEMORIA:

6

CICLOS DE RELOJ:

23

EJEMPLO:

RLC (IX+2)

Contenido del registro índice «IX»

(IX):	1 1 1 1 0 0 0 0	F0h
	0 1 0 1 1 0 1 0	5Ah

Contenido del octeto de memoria F05Ch

F05Ch:	1 1 1 1 1 1 1 0	FEh

Instrucción

RLC (IX+2):	1 1 0 1 1 1 0 1	DD
	1 1 0 0 1 0 1 0	CBh
	0 0 0 0 0 0 1 0	02h
	0 0 0 0 0 1 1 0	06h

Contenido del octeto de memoria F05Ch después de la ejecución

F05CH:	1 1 1 1 1 1 0 1	FDh

Indicadores de condición después de la ejecución

S	Z		H		P/V	N	C
1	0	x	0	x	0	0	1

RLC (IY + d)

OBJETO:

Rota a la izquierda, un bit, el octeto de memoria direccionado por el contenido del registro índice «IY» más el entero de desplazamiento «d», el cual puede adquirir los valores desde —128 a +127. El valor del bit 7 saliente se copia en el bit 0 entrante y en el indicador de acarreo «C». Ver Figura 9-1.

CODIGO DE MAQUINA:

1 1 1 1 1 1 0 1	FDh
1 1 0 0 1 0 1 1	CBh
←— d —→	
0 0 0 0 0 1 1 0	06h

INDICADORES DE CONDICION QUE AFECTA:

S; pone 1 - si el resultado es negativo;
pone 0 - en cualquier otro caso
Z; pone 1 - si el resultado es cero;
pone 0 - en cualquier otro caso
H; pone 0 - siempre
N; pone 0 - siempre
C; pone el valor que tenía el bit 7 del octeto antes de la ejecución
P/V; pone 1 - si la paridad es par;
pone 0 - en cualquier otro caso

CICLOS DE MEMORIA:

6

CICLOS DE RELOJ:

23

EJEMPLO:

RLC (IY-2)

Contenido del registro índice «IY»

(IY):	1 1 0 0 1 1 0 0	CCh
	0 0 1 1 0 1 1 0	36h

Contenido del octeto de memoria CC34h

CC34h:	0 0 1 1 0 0 1 1	33h

Instrucción

RLC (IY-2):	1 1 0 1 1 1 0 1	DD
	1 1 0 0 1 0 1 0	CBh
	1 1 1 1 1 1 1 0	FEh
	0 0 0 0 0 1 1 0	06h

Contenido del octeto de memoria CC34h después de la ejecución

CC34h:	0 1 1 0 0 1 1 0	66h

Indicadores de condición después de la ejecución

S	Z		H		P/V	N	C
0	0	x	0	x	1	0	0

RL r

OBJETO:

Rota a la izquierda, un bit, el contenido del registro representado por «r». El contenido del bit 7 saliente se copia en el indicador de acarreo «C», y el valor anterior del indicador de acarreo se copia

en el bit Ø entrante. Ver Figura 9-2. El código de representación de «r» es el señalado más abajo.

Registro	Código
B	ØØØ
C	ØØ1
D	Ø1Ø
E	Ø11
H	1ØØ
L	1Ø1
A	111

CODIGO DE MAQUINA:

1 1 Ø Ø 1 Ø 1 1	CBh
Ø Ø Ø 1 Ø ←r→	

INDICADORES DE CONDICION QUE AFECTA:

S; pone 1 - si el resultado es negativo;
pone Ø - en cualquier otro caso
Z; pone 1 - si el resultado es cero;
pone Ø - en cualquier otro caso
H; pone Ø - siempre
N; pone Ø - siempre
C; pone el valor que tenía el bit 7 del registro r antes de la ejecución
P/V; pone 1 - si la paridad es par;
pone Ø - en cualquier otro caso

CICLOS DE MEMORIA:

2

CICLOS DE RELOJ:

8

EJEMPLO:

RL H

Contenido del registro «H»

(H): Ø Ø Ø Ø Ø Ø 1 Ø Ø2h

INSTRUCCIONES DE ROTACION

Código Fuente	Hexadecimal	Decimal
RL A	17	23
RL B	CB,1Ø	2Ø3,16
RL C	CB,11	2Ø3,17
RL D	CB,12	2Ø3,18
RL E	CB,13	2Ø3,19
RL H	CB,14	2Ø3,2Ø
RL L	CB,15	2Ø3,21
RL (HL)	CB,16	2Ø3,22
RL (IX+d)	DD,CB,d,16	221,2Ø3,d,22
RL (IY+d)	FD,CB,d,16	253,2Ø3,d,22
RLC A	CB,Ø7	2Ø3,7
RLC B	CB,ØØ	2Ø3,Ø
RLC C	CB,Ø1	2Ø3,1
RLC D	CB,Ø2	2Ø3,2
RLC E	CB,Ø3	2Ø3,3
RLC H	CB,Ø4	2Ø3,4
RLC L	CB,Ø5	2Ø3,5
RLC (HL)	CB,Ø6	2Ø3,6
RLC (IX+d)	DD,CB,d,Ø6	221,2Ø3,d,6
RLC (IY+d)	FD,CB,d,Ø6	253,2Ø3,d,6
RR A	1F	31
RR B	CB,18	2Ø3,24
RR C	CB,19	2Ø3,25
RR D	CB,1A	2Ø3,26
RR E	CB,1B	2Ø3,27
RR H	CB,1C	2Ø3,28
RR L	CB,1D	2Ø3,29
RR (HL)	CB,1E	2Ø3,3Ø
RR (IX+d)	DD,CB,d,1E	221,2Ø3,d,3Ø
RR (IY+d)	FD,CB,d,1E	253,2Ø3,d,3Ø
RRC A	CB,ØF	2Ø3,15
RRC B	CB,Ø8	2Ø3,8
RRC C	CB,Ø9	2Ø3,9
RRC D	CB,ØA	2Ø3,1Ø
RRC E	CB,ØB	2Ø3,11
RRC H	CB,ØC	2Ø3,12
RRC L	CB,ØD	2Ø3,13
RRC (HL)	CB,ØE	2Ø3,14
RRC (IX+d)	DD,CB,d,ØE	221,2Ø3,d,14
RRC (IY+d)	FD,CB,d,ØE	253,2Ø3,d,14

Fig. 9-5. Tabla de codificación para instrucciones de rotación.

Indicador de acarreo C = Ø

Instrucción

RL H:	1 1 Ø Ø 1 Ø 1 Ø	CBh
	Ø Ø Ø 1 Ø 1 Ø Ø	14h

Contenido del registro «H» después de la ejecución

(H): Ø Ø Ø Ø Ø 1 Ø Ø Ø4h

Indicadores de condición después de la ejecución

S	Z		H		P/V	N	C
Ø	Ø	x	Ø	x	Ø	Ø	Ø

RL (HL)

OBJETO:
Rota a la izquierda, un bit, el octeto de memoria direccionado por el contenido del par de registros «HL». El contenido del bit 7 saliente se copia en el indicador de acarreo «C» y el indicador de acarreo anterior se copia en el bit Ø entrante. Ver Figura 9-2.

CODIGO DE MAQUINA:

1 1 Ø Ø 1 Ø 1 1	CBh
Ø Ø Ø 1 Ø 1 1 Ø	16h

INDICADORES DE CONDICION QUE AFECTA:

S; pone 1 - si el resultado es negativo;
pone Ø - en cualquier otro caso
Z; pone 1 - si el resultado es cero;
pone Ø - en cualquier otro caso
H; pone Ø - siempre
N; pone Ø - siempre
C; pone el valor que tenía el bit 7 del octeto antes de la ejecución
P/V; pone 1 - si la paridad es par;
pone Ø - en cualquier otro caso

CICLOS DE MEMORIA:

4

CICLOS DE RELOJ:

15

EJEMPLO:

RL (HL)

Contenido del par de registros «HL»

INSTRUCCIONES DE ROTACION

NEMONICO	S	Z	x	H	x	P/V	N	C	No. DE BYTES	CICLOS MEM.	CICLOS REL.
RLC A	.	.	x	Ø	x	.	Ø	↕	1	1	4
RLC r	↕	↕	x	Ø	x	P	Ø	↕	2	2	8
RLC (HL)	↕	↕	x	Ø	x	P	Ø	↕	2	4	15
RLC (IX+d)	↕	↕	x	Ø	x	P	Ø	↕	4	6	23
RLC (IY+d)	↕	↕	x	Ø	x	P	Ø	↕	4	6	23
RL A	.	.	x	Ø	x	.	Ø	↕			
RL s	↕	↕	x	Ø	x	P	Ø	↕			
RRC A	.	.	x	Ø	x	.	Ø	↕			
RRC s	↕	↕	x	Ø	x	P	Ø	↕			
RR A	.	.	x	Ø	x	.	Ø	↕			
RR s	↕	↕	x	Ø	x	P	Ø	↕			

NOTAS:

1.- Los signos tienen el siguiente significado:
"↕": El indicador cambia de valor de acuerdo con el resultado de la instrucción.
"x": El bit adquiere un estado indeterminado.
".": El indicador no es afectado por la instrucción y conserva su anterior contenido.
"Ø": El indicador se pone siempre a "cero".
"P": El indicador "P/V" actua como indicador de paridad.

2.- La letra "r" indica cualquiera de los registros: "B", "C", "D", "E", "H" ó "L".

3.- La letra "s" indica cualquiera de los operandos: "r", "(HL)", "(IX+d)" ó "(IY+d)".

Fig. 9-6. Tabla resumida de indicadores y ciclos para las instrucciones de rotación.

Expansión

JOYSTICK «INVESTICK»



Detalle de los pulsadores de disparos y sus contactos.



Pulsadores a ambos lados. Un detalle con los zurdos.



El «investick» destaca por su robustez y su facilidad de manejo gracias a la fijación de la base.

Desde el mes de mayo pasado, la empresa F.M. y Asociados, S. A., fabrica el primer Joystick español y del que tiene la exclusiva de comercialización para el mercado nacional la firma Investrónica, S. A.

Este nuevo periférico, conocido con el nombre de Investick es de robusta construcción, posee una empuñadura ergonómica y dos disparadores, para los dedos pulgar e índice.

En la base cuenta con otros dos más con objeto de que pueda servir para cualquiera de las manos (un detalle con los zurdos). Se puede usar perfectamente sin estar apoyado en la mesa, gracias a unas estrías que impiden que pueda deslizarse involuntariamente de entre las manos. Sin embargo, la forma más apropiada y eficaz de utilización es apoyándolo en una superficie lisa y sólida, mediante las cuatro ventosas de fijación que tiene en su base.

Su mecanismo es sumamente fiable y está compuesto por un circuito impreso provisto de contactos de acero templado, sujetos con un material de poliéster antioxidante. La unión de la empuñadura con la base se efectúa mediante una semiesfera interior de nylon capaz de soportar una presión aproximada de 400 kilos, por lo que su rotura es prácticamente imposible. Lástima que esta robustez general no haya sido complementada con una placa de circuito impreso en fibra de vidrio, en vez de ser de bakelita, un material bastante más frágil.

El producto está plenamente garantizado contra cualquier defecto de fabricación y la empresa que lo ha desarrollado asegura que durante la campaña de navidades se han comercializado alrededor de 35.000 unidades.

BATCODE: UNA AUTENTICA BATALLA DENTRO DE TU ORDENADOR

Sergio MARTINEZ LARA

El interior de nuestro ordenador es como una enorme ciudad construida por celdillas de memoria. En cada una de estas supuestas casas habitan pacíficos Bytes. Pero... ¿qué ocurre si un buen día deciden pelearse entre ellos intentando conquistar la ciudad? Más aún, ¿qué pasa si quedan divididos en dos poderosos ejércitos, uno de los cuales está bajo tu mando y el otro, a las órdenes de cualquiera de tus amigos?: una auténtica guerra se ha desencadenado dentro de tu Spectrum...



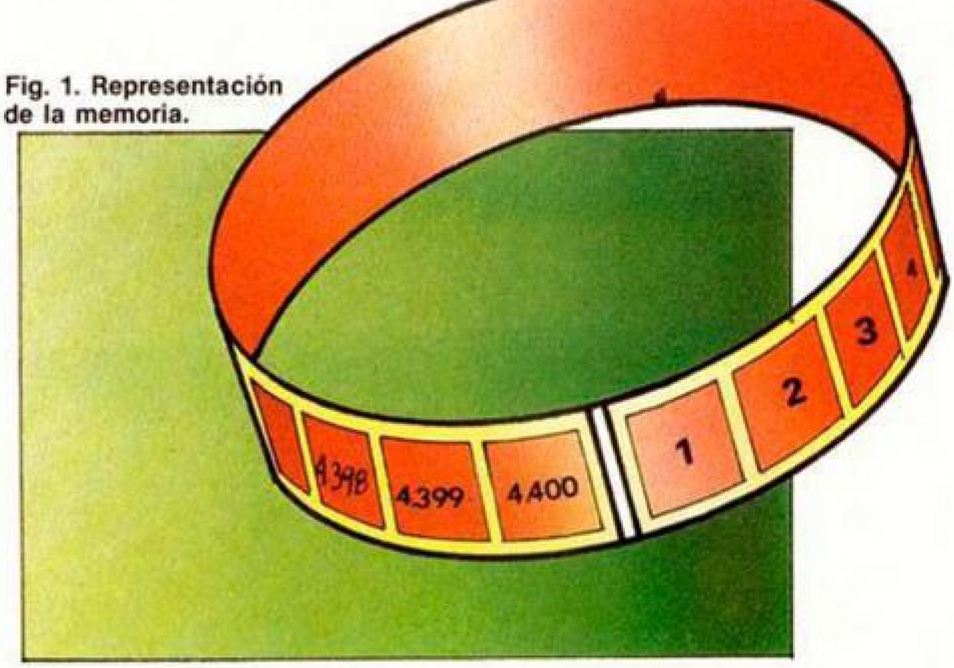
Tal vez hayas oído alguna historia sobre los «Hackers». El diccionario de lengua inglesa los define como «personas que son capaces de abrirse paso, a cualquier precio, a través de las barreras o protecciones que les impone un determinado sistema informático o de cualquier otro tipo». La historia suele ser siempre similar: un joven o un grupo de ellos, cuyas edades generalmente no sobrepasan los 17 ó 18 años, consiguen penetrar en un sistema complejo de ordenadores de alguna entidad importante (léase la NASA, el Pentágono u otro organismo parecido) poniendo en peligro la seguridad y el secreto de importantes datos; todo ello con un simple ordenador personal conectado a la red telefónica mediante un «modem». En fin, todos conocéis la película «Juegos de guerra».

También circulan en los ambientes informáticos otro tipo de historias basadas en la misma filosofía. Un programador desaprensivo que ha sido despedido de su empresa, escribe un programa que tiene la capacidad de reproducirse y multiplicarse, asfixiando al complejo sistema en el que ha sido introducido de forma que, al cabo de poco tiempo, dicho sistema es colapsado. Entonces aparece el programador «bueno» de esta película, que escribe un programa capaz de ir destruyendo al anterior y cuando ha terminado con él, destruirse a sí mismo.

Y aquí tenemos un buen argumento para un juego. El juego fue sugerido por A. K. Dewdney en la revista de divulgación científica norteamericana «Scientific American» hace aproximadamente dos años. La idea es en sí sencilla y original. Supongamos dos programas que comparten un «hábitat» común, es decir, la memoria de un ordenador. Ellos deben luchar entre sí por su supervivencia y a la vez intentar aniquilar al otro. Esto, naturalmente, no es tarea fácil ya que ninguno de estos programas conoce dónde se encuentra exactamente el otro, ni siquiera, como



Fig. 1. Representación de la memoria.



se explicará, sabe cuál es su posición dentro de la memoria central. El programa debe averiguar dónde se encuentra su oponente, atacarlo y todo esto sin olvidar su propia defensa.

El lenguaje «BATCODE»

¿Cómo se puede realizar esto? Tenemos, en primer lugar, que definir un lenguaje de programación especial que permita, con sencillas «órdenes», controlar apropiadamente nuestro «ejército» de bytes. A este lenguaje le hemos llamado «BATCODE».

Es bastante parecido a un lenguaje de tipo ensamblador y tal vez este juego pueda servir también como introducción al Assembler, ya que la estructura y funcionamiento son prácticamente idénticas.

Los programas escritos en BATCODE se ejecutan en una matriz de memoria con 4400 posiciones. Estas están numeradas de forma cíclica, es decir, la secuencia que se sigue es por ejemplo: 4397, 4398, 4399, 4400, 1,2, ... etc. Como se ve, la posición siguiente a la 4400 es la posición número 1. Nos podríamos imaginar este campo de batalla como un cilindro (ver figura 1).

Las instrucciones disponibles nos permiten mover contenidos de memoria de un lado a otro, sumar, restar, comparar, etc..., como se muestra en el repertorio completo de instrucciones de la tabla 1.

Un juego para dos contrincantes

Como has podido suponer, éste es un juego para dos personas, cada una de las cuales escribirá un programa. Luego, estos programas se colocarán en la memoria aleatoriamente, separados entre sí por, al menos, mil posiciones, y entonces se empieza a ejecutar cada uno de ellos, de forma que primero se ejecuta una instrucción de un programa y luego otra instrucción del segundo. Y así sucesivamente hasta que llega un momento en el que no se puede ejecutar una determinada instrucción. El programa al que corresponda esa instrucción resulta ser el perdedor.

Vamos ahora con el programa encargado de realizar ésta tarea, una especie de árbitro imparcial que actúa cediendo el turno alternativamente a cada uno de los programas. En primer lugar, hay que copiar cuidadosamente el listado Basic del programa EDITOR DE BATCODE y una vez terminado, salvarlo en una cinta de cassette con SAVE «BATCODE» LINE 9000. Con esto conseguiremos que el programa se autoejecute a partir de la línea 9000 una vez que se cargue.

TABLA 1

Cod.	Instrucción	Descripción
0	DAT X	Declaración de dato
1	SAL X	Salto a X
2	DSC X Y	Decrementa y salta si es cero
3	MOV X Y	Mueve X a Y
4	SUM X Y	Suma a X el valor Y
5	RES X Y	Resta a X el valor Y
6	SAC X Y	Salta a X si Y es cero
7	SMY X Y	Salta a X si Y es mayor que cero
8	CMP X Y	Compara X e Y entre sí
9	COD X Y	Coloca el código de X en Y
10	SET X Y	Pon el valor X en la posición Y
11	INT X	Intercambiar los operandos de X
12	NOP	No operación

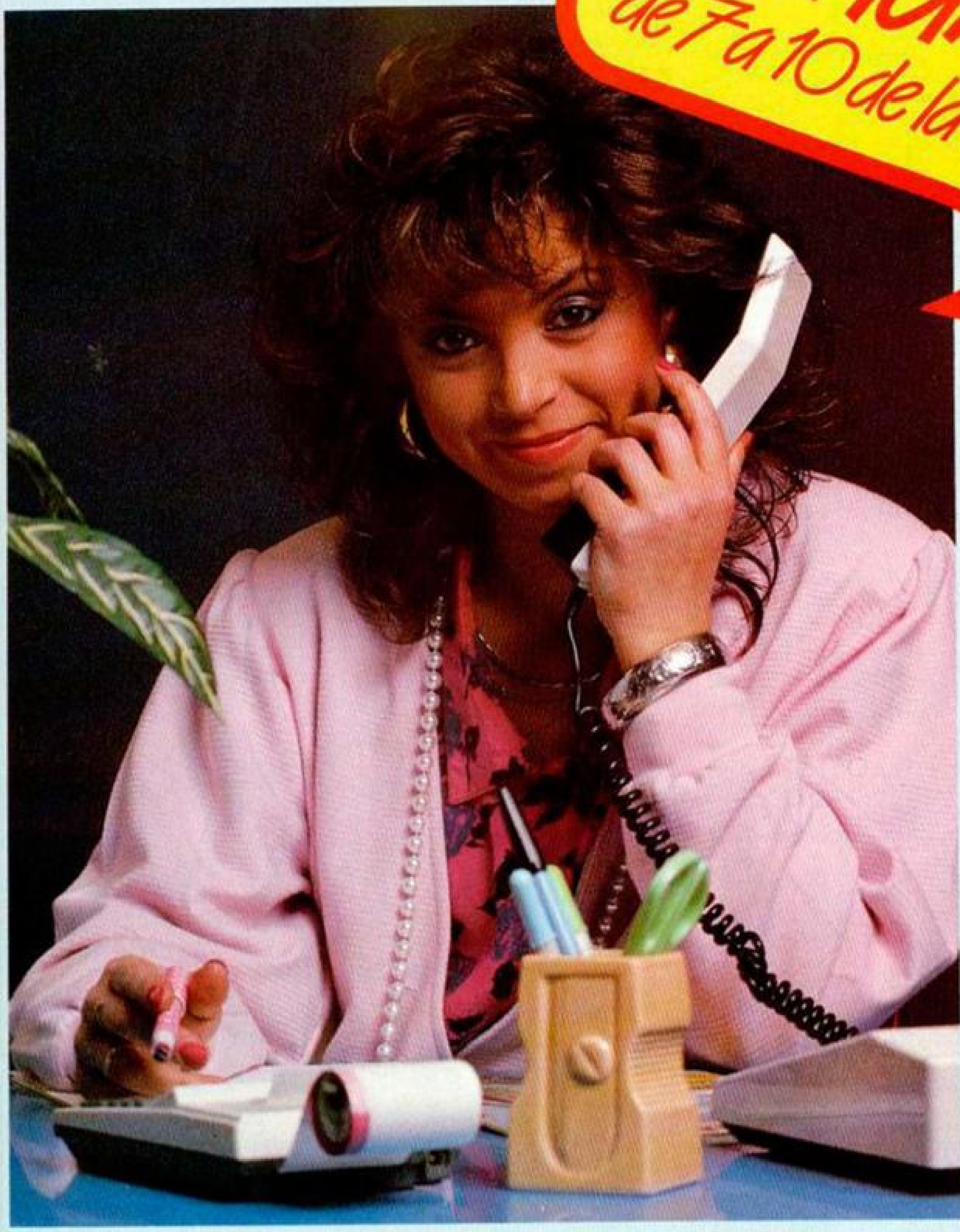
LISTADO BASIC DEL EDITOR DE «BATCODE»

```
  10 REM
  20 REM         © 1986
  30 REM   Sergio Martinez Lara
  40 REM
  50 DEF FN j$(x)="000"( TO 3-LEN STR$ x)+STR$ x
  60 DEF FN s$(x)="0000"( TO 4-LEN STR$ x)+STR$ x
  70 DEF FN m(x)=INT (x/256)
  80 DEF FN r(x)=x-256*INT (x/256)
  90 DEF FN p(x)=PEEK x+256*PEEK (x+1)
 100 DATA "32495756543203698271737932776582847378699032"
 110 DATA "DAT","SAL","DSC","MOV","SUM","RES","SAC","SMY","CMP","COD","SET","INT","NOP"
 120 RESTORE
 130 READ w$: LET c$="  ": LET c$(2)=CHR$ 127
 140 FOR i=1 TO LEN w$ STEP 2: LET c$=c$+CHR$ VAL w$(i TO i+1): NEXT i
 150 PAPER 1: INK 7: BORDER 1: CLS
 160 PRINT AT 0,7; PAPER 2;"* EDITOR DE BATCODE *"
 170 PRINT AT 2,5; INVERSE 1; BRIGHT 1;c$
 180 DIM T$(39): DIM f$(22000): DIM p$(2500): DIM m$(2500)
 190 LET linea=-1
 200 LET s$="                                "
 210 FOR I=1 TO 39 STEP 3: READ t$(i TO i+2): NEXT i
 220 LET p$(1)=p$(1): RANDOMIZE USR 65502
 230 LET m$(1)=m$(1): RANDOMIZE USR 65502
 240 PAUSE 50: CLS
 250 GO SUB 1000: INPUT "Desea finalizar el juego (S/N) "; LINE l$: IF l$="S" THEN STOP
 260 GO TO 250
1000 REM analizador sintactico
1010 POKE 23658,8: INPUT LINE l$
1020 IF l$="" THEN RETURN
1030 LET i=1: GO SUB 1830: LET l$=l$(i TO ): LET i=1: IF error THEN GO TO 1000
1040 LET long=LEN l$: LET l$=l$+"  "
1050 IF l$(1)="L" THEN LET Z$="L": GO TO 2040
1060 IF l$(1)="D" THEN LET Z$="D": GO TO 2040
1070 IF l$(1)="P" THEN LET z$="L": GO TO 2010
1080 IF l$(1)="I" THEN LET z$="D": GO TO 2010
1090 IF l$(1)="S" THEN GO TO 2470
1100 IF l$(1)="C" THEN GO TO 2580
1110 IF l$(1)="N" THEN GO TO 2650
1120 IF l$(1)="A" THEN GO TO 2720
1130 IF l$(1)="T" THEN GO TO 2850
1140 IF l$(1)="E" THEN GO TO 3130
1150 IF l$(1)="B" THEN GO TO 3000
1160 IF l$(1)="M" THEN GO TO 3060
1170 LET i=1: GO SUB 1640
1180 IF error THEN GO SUB 1890: GO TO 1010
1190 LET dir=valor
1200 IF dir<0 OR dir>500 THEN GO SUB 1890: GO TO 1010
1210 LET r$=l$(i TO )
1220 IF long<=4 THEN GO TO 1980
1230 IF l$(i)<>" " THEN GO SUB 1890: GO TO 1010
1240 GO SUB 1830
1250 IF long<i+3 THEN LET i=long: GO SUB 1890: GO TO 1010
1260 LET w$=l$(i TO i+2): LET i=i+3
1270 IF long<i THEN GO TO 1290
1280 IF l$(i)<>" " THEN GO SUB 1890: GO TO 1010
1290 FOR j=0 TO 12
1300 IF w$=t$(j*3+1 TO j*3+3) THEN GO TO 1330
1310 NEXT j
1320 GO SUB 1890: GO TO 1010
1330 LET cod=j
1340 IF (cod=0 OR cod=1 OR cod=11) THEN GO SUB 1580: GO TO 1360
1350 GO SUB 1490
1360 IF error THEN GO SUB 1890: GO TO 1010
1370 LET x=op1
1380 GO SUB 1920
1390 LET v$=w$
1400 IF cod=0 OR cod=1 THEN LET w$=CHR$ 0+CHR$ 0: LET md2=0: GO TO 1430
1410 LET x=op2
1420 GO SUB 1920
1430 LET p$(dir*5-4 TO dir*5)=CHR$ (16*cod+4*md1+md2)+v$+w$
1440 PRINT AT 20,0;s$;s$
1450 IF linea<19 THEN : LET linea=linea+1: GO TO 1470
1460 PRINT AT 21,0: POKE 23692,255: PRINT
1470 PRINT AT linea,0;FN j$(dir);r$
1480 GO TO 1000
1490 REM analisis de 2 operandos
1500 GO SUB 1730
1510 IF error THEN RETURN
1520 IF I>long THEN LET error=1: RETURN
1530 LET md1=md: LET op1=valor
1540 GO SUB 1730
1550 IF error THEN RETURN
1560 LET md2=md: LET op2=valor
1570 RETURN
1580 REM analisis de 1 operando
1590 GO SUB 1730
1600 IF error THEN RETURN
1610 LET md1=md: LET op1=valor
1620 RETURN
1630 REM deteccion de un numero
1640 GO SUB 1830: IF error THEN GO SUB 1890: GO TO 1000
1650 LET w$=""
1660 FOR i=i TO i+4
1670 IF l$(i)=" " THEN GO TO 1710
1680 IF (l$(i)<"0" OR l$(i)>"9") AND l$(i)<>"+" AND l$(i)<>"-" THEN LET error=1: RETURN
1690 LET w$=w$+l$(i)
1700 NEXT i
1710 LET valor=VAL w$
1720 RETURN
1730 REM numero con modificador
1740 LET md=0
1750 IF i>long THEN LET error=1: RETURN
1760 IF l$(i)<>" " THEN LET error=1: RETURN
1770 LET i=i+1
1780 IF i>long THEN LET error=1: RETURN
1790 IF l$(i)="#" THEN LET i=i+1: LET md=1: GO TO 1810
1800 IF l$(i)="@" THEN LET i=i+1: LET md=2: GO TO 1810
1810 GO SUB 1630
1820 RETURN
1830 REM salta espacios
1840 LET error=0
1850 IF i>LEN l$ THEN LET error=1: RETURN
1860 IF l$(i)=" " THEN LET i=i+1: GO TO 1850
1870 RETURN
1880 REM mensaje de error
1890 PRINT AT 20,0;s$;s$;AT 20,0;l$;AT 21,i-1;"↑ ERROR"
1900 BEEP 0.5,20
1910 RETURN
1920 REM conversor complemento a 2
1930 IF x<0 THEN LET x=x+65536
1940 LET w$="  "
1950 LET w$(2)=CHR$ (INT (x/256))
1960 LET w$(1)=CHR$ (x-256*CODE w$(2))
1970 RETURN
1980 REM borrado de una linea
1990 LET cod=12: LET op1=0: LET op2=op1: LET md1=op1: LET md2=op1
2000 GO TO 1370
2010 REM insercion
2020 LET x=3
2030 GO TO 2060
2040 REM listado
2050 LET x=2
2060 CLS
2070 LET dir=1
2080 IF LEN l$<3 THEN GO TO 2120
2090 LET i=2: GO SUB 1630
2100 IF error=0 AND valor>1 AND valor<=500 THEN LET dir=valor*5-4
2110 IF z$="D" AND (error=0 AND valor>1 AND valor<=4400) THEN LET dir=valor*5-4
2120 FOR i=0 TO 19
2130 IF Z$="D" THEN LET w$=F$(dir TO dir+4): GO TO 2150
2140 LET w$=p$(dir TO dir+4)
2150 GO SUB 2360
2160 IF Z$<>"D" THEN PRINT #x;FN j$(INT (dir/5)+1);: GO TO 2180
2170 PRINT #x;FN s$(INT (dir/5)+1);
2180 PRINT #x;" ";t$(cod*3+1 TO cod*3+3);" ";
2190 IF cod=12 THEN GO TO 2270
2200 IF md1=1 THEN PRINT #x;"#";: GO TO 2220
2210 IF md1=2 THEN PRINT #x;"@";: GO TO 2220
2215 PRINT #x;" ";
2220 PRINT #x;op1;TAB 14;
2230 IF cod<=1 OR cod=11 THEN GO TO 2270
2240 IF md2=1 THEN PRINT #x;CHR$ (8);"#";
2250 IF md2=2 THEN PRINT #x;CHR$ (8);"@";
2260 PRINT #x;op2;
2270 LET dir=dir+5: IF dir>22000 THEN LET dir=1
2280 IF Z$="L" AND dir>2500 THEN LET dir=1
2290 PRINT #x: NEXT i
2300 PRINT AT 21,0;"Pulse 'S' para seguir listando"
2310 PAUSE 0: LET w$=INKEY$
2320 IF w$="s" OR w$="S" THEN CLS : GO TO 2120
2330 LET linea=19
2340 PRINT AT 20,0;s$;s$
2350 GO TO 1000
2360 REM decodificador
2370 LET md2=CODE w$(1)
2380 LET op1=CODE w$(2)+256*CODE w$(3)
2390 IF op1>32767 THEN LET op1=op1-65536
2400 LET op2=CODE w$(4)+256*CODE w$(5)
2410 IF op2>32767 THEN LET op2=op2-65536
2420 LET cod=INT (md2/16)
2430 LET md2=md2-cod*16
2440 LET md1=INT (md2/4)
2450 LET md2=md2-md1*4
2460 RETURN
2470 REM salvar programa
2480 CLS
2490 PRINT AT 0,7; PAPER 2;"* SALVAR PROGRAMAS *"
2500 PRINT AT 2,5; INVERSE 1; BRIGHT 1;c$
2510 INPUT "Nombre del programa a salvar: "; LINE w$
2520 IF w$="" THEN GO TO 2840
2530 SAVE w$ DATA p$()
2540 PRINT #1;"Rebobina la cinta para verificar"
2550 PRINT AT 10,0;
2560 VERIFY w$ DATA p$()
2570 GO TO 2840
2580 REM Carga de programa
2590 CLS
2600 PRINT AT 0,7; PAPER 2;"* CARGAR PROGRAMAS *"
2610 PRINT AT 2,5; INVERSE 1; BRIGHT 1;c$
2620 INPUT "Nombre del programa a cargar: "; LINE w$
2630 LOAD "" DATA p$()
2640 GO TO 2840
2650 REM NUEVO PROGRAMA
2660 CLS
2670 PRINT AT 0,12; FLASH 1; PAPER 2;"* ""NEU"" *"
2680 PRINT AT 2,5; INVERSE 1; BRIGHT 1;c$
2690 INPUT "Seguro (S/N)";w$
2700 IF w$="S" OR w$="s" THEN LET p$(1)=p$(1): RANDOMIZE USR 65502: GO TO 2840
2710 GO TO 2840
2720 REM Pantalla de ayuda
2730 LET w$="A-Ayuda     B-Borr. lin.C-Carga prg.S-Salva prg.M-Mover lin.N-Nuevo prg.L-Listar P-Imp. list.D-Dump mem. I-Imp. Dump T-TransladarE-Ejecucion "
2740 PRINT AT 3,17; PAPER 6;"               "
2750 PRINT AT 4,17; PAPER 6;" ";AT 4,31;" "
2760 FOR i=0 TO 11
2770 PRINT AT i+5,17; PAPER 6;" "; PAPER 1;" ";w$(i*12+1 TO (i+1)*12); PAPER 6;AT i+5,31;" "
2780 NEXT i
2790 PRINT AT 17,17; PAPER 6;" ";AT 17,31;" "
2800 PRINT AT 18,17; PAPER 6;"               "
2810 PRINT AT 20,7; PAPER 2;"* PANTALLA DE AYUDA *"
2820 PRINT #1;AT 0,5; INVERSE 1; BRIGHT 1;c$
2830 PAUSE 0
2840 INK 7: PAPER 1: BORDER 1: CLS : LET linea=-1: GO TO 1000
2850 REM TRANSLADAR PROGRAMA
2860 CLS
2870 PRINT AT 0,6; PAPER 2;"* TRANSLADAR PROGRAMA *"
2880 PRINT AT 2,5; INVERSE 1; BRIGHT 1;c$
2890 PRINT AT 9,7;"S .- Salvar prog."
2900 PRINT AT 11,7;"C .- Cargar prog."
2910 PRINT AT 13,7;"I .- Inter. prgs."
2920 PRINT #0;" Elija una opcion (S/C/I) ";: PAUSE 0: LET o$=INKEY$: PRINT #0;o$: IF CODE o$>96 THEN LET o$=CHR$ (CODE o$-32)
2930 IF o$="S" THEN LET m$=p$
2940 IF o$="C" THEN LET p$=m$
2950 IF o$<>"I" THEN GO TO 2840
2960 LET f$( TO 2500)=m$
2970 LET m$=p$
2980 LET p$=f$( TO 2500)
2990 GO TO 2840
3000 REM borrado de lineas
3010 LET i=2: GO SUB 1490
3020 IF error THEN GO SUB 1890: GO TO 1000
3030 IF (op1+op2>501) OR op1<1 OR op2<1 THEN GO SUB 1890: GO TO 1000
3040 LET op1=op1*5-4: POKE 65498,FN r(op2): POKE 65499,FN m(op2)
3050 LET p$(op1)=p$(op1): RANDOMIZE USR 65497: GO TO 1000
3060 REM mover lineas
3070 LET i=2: GO SUB 1490: IF error THEN GO SUB 1880: GO TO 1000
3080 GO SUB 1630: IF error THEN GO SUB 1880: GO TO 1000
3090 IF op1+valor>500 OR op2+valor>500 OR valor<0 OR op1<1 OR op2<1 THEN GO SUB 1880: GO TO 1000
3100 LET op1=op1*5-4: LET op2=op2*5-4: LET valor=valor*5-1
3110 LET m$=p$: LET p$(op2 TO op2+valor)=m$(op1 TO op1+valor)
3120 LET m$(1)=m$(1): RANDOMIZE USR 65502: GO TO 1000
3130 REM [illegible]
3140 CLS : LET pos2=INT (RND*2400+1000): POKE 64259,FN r(pos2): POKE 64260,FN m(pos2)
3150 PRINT AT 0,5; PAPER 2;"* BATALLA DE PROGRAMAS *"
3160 PRINT AT 2,5; INVERSE 1; BRIGHT 1;c$
3170 INPUT "Ejecucion paso a paso (S/N) "; LINE o$: POKE 64064,o$="s" OR o$="S"
3180 LET f$(1)=f$(1): RANDOMIZE USR 65507
3190 LET f$(1 TO 2500)=p$: LET f$(pos2*5+1 TO pos2*5+2500)=m$
3200 INK 0: PAPER 6: BORDER 6
3210 LET f$(1)=f$(1): RANDOMIZE USR 64236
3220 IF o$="s" OR o$="S" THEN PRINT AT 20,6;"   Pulse una tecla   ": PAUSE 0: PRINT AT 20,6; PAPER 2;"                    "
3230 IF INKEY$="" THEN BEEP .1,.1: PRINT AT 20,6; FLASH 1;" Ejecucion interrumpida ": GO TO 3260
3240 IF PEEK 65122=0 THEN PRINT AT 20,6; FLASH 1;"   Pierde el programa ";CHR$ (64+PEEK 65121);"   ": GO TO 2830
3250 LET f$(1)=f$(1): RANDOMIZE USR 64444: GO TO 3220
3260 PAUSE 0: IF INKEY$="c" OR INKEY$="C" THEN PRINT AT 20,6; PAPER 2;"                    ": GO TO 3240
3270 GO TO 2840
9000 REM CARGA C.M.
9010 CLEAR 63999: PRINT AT 10,5; FLASH 1;"CARGA DE CODIGO MAQUINA"
9020 LOAD ""CODE 64000
9040 GO TO 10
```

Este programa no está, ni mucho menos, completo. Le falta un importante bloque de Código Máquina que por razones de espacio publicaremos en el próximo número. Un poco de paciencia.

Si eres lector habitual de esta revista

¡llámame!
de 7 a 10 de la tarde



Te estoy esperando.

Tengo muchas cosas que contarte... y muy interesantes. De momento, te propongo la posibilidad de *AHORRAR 1.600* ptas. y, además, con un poco de suerte, *GANAR UNA VESPINO* ¿Qué te parece? Pues esto es sólo un avance. Cuando me llames te contaré más cosas que seguro te gustarán.

Pero no te demores, porque a una mujer nunca se le hace esperar. Tienes de plazo hasta el 31 de marzo. Después, habrás perdido tu oportunidad.

(91) 654 32 11

HOBBY PRESS, S.A.

LOS JUSTICIEROS DEL SOFTWARE

CAMELOT WARRIORS

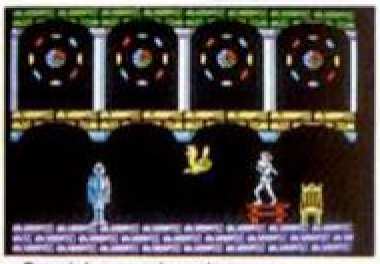
Con este programa de Dinamic se despiden de nuestras páginas estos ocho justicieros que, fielmente, cada semana, han colaborado con nosotros en la difícil tarea de ser ecuánimes y objetivos en la crítica de un programa. Desde aquí, os damos las gracias.

«Tiene efectos cómicos»

● **POSITIVO.** La principal virtud del programa son sus fantásticos gráficos. La pantalla tiene un gran colorido y son totalmente diferentes entre sí. El juego es bastante original y tiene efectos cómicos.

● **NEGATIVO.** Los movimientos no son demasiado buenos, resultando algo artificiales. En cuanto al sonido sin duda lo podrían haber hecho mejor.
Puntuación: 8.

JAVIER REDONDO



«Sonido casi nulo»

● **POSITIVO.** El movimiento es excelente y los gráficos muy buenos. Muy original y con una presentación genial.

● **NEGATIVO.** El sonido casi no existe.
Puntuación: 9.

JUSTO SORIA

«El Summun»

● **POSITIVO.** El 99 por 100 de este juego es positivo, ya que tiene unos gráficos perfectos, un movimiento simpático, una presentación colosal y una gran originalidad y adicción. No hay palabras en el mundo para nombrar este juego.

● **NEGATIVO.** Lo único que se le puede pedir es algún efecto sonoro más.
Puntuación: 10.

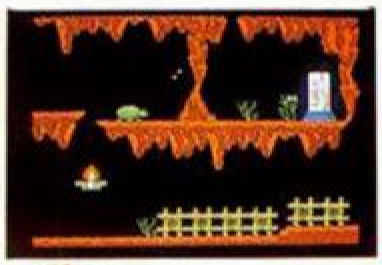
GORKA POLITE

«La melodía es genial»

● **POSITIVO.** El sonido es el apropiado. El decorado es excepcional y la melodía es genial.

● **NEGATIVO.** La animación es escasa, el gráfico del personaje, es muy simple. Nivel de dificultad muy elevado.
Puntuación: 9.

CELIA MENENDEZ

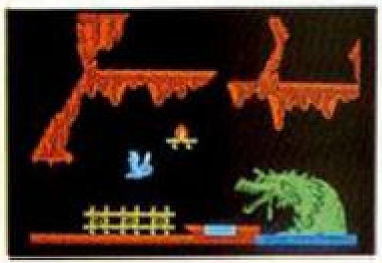


«Poco adictivo»

● **POSITIVO.** Los gráficos son bastante aceptables. Los movimientos son excelentes. El sonido está logrado durante todo el juego. Es bastante original. La pantalla de presentación está muy bien realizada.

● **NEGATIVO.** El juego es bastante difícil, no es muy adictivo, porque no logramos pasar muchas pantallas.
Puntuación: 9.

DANIEL GONZALEZ



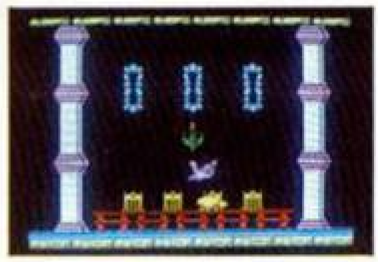
«Un programa original»

● **POSITIVO.** Es un programa estupendo, tiene una buenísima calidad sobre todo en el movimiento del personaje. El colorido ha sido muy bien tratado y decorados bastante logrados. La adicción es muy alta. Un programa original.

● **NEGATIVO.** Excesiva dificultad y no se han esforzado mucho en el sonido.
Puntuación: 9.

PURIFICACION HERNANDEZ





«Gráficos excelentes»

● **POSITIVO.** Los gráficos son excelentes, así como el colorido de los mismos y el movimiento del personaje. La pantalla de presentación es original.

● **NEGATIVO.** El nivel de dificultad es alto, y el sonido es casi nulo.
Puntuación: 9.

JOSE CARLOS NUÑEZ

«Alta dificultad»

● **POSITIVO.** Gráficos maravillosos. El movimiento perfecto aunque un tanto lento. Sonido preciso y muy bueno. La ambientación está muy conseguida.

● **NEGATIVO.** La dificultad es muy alta.
Puntuación: 9.

JAVIER VALERO



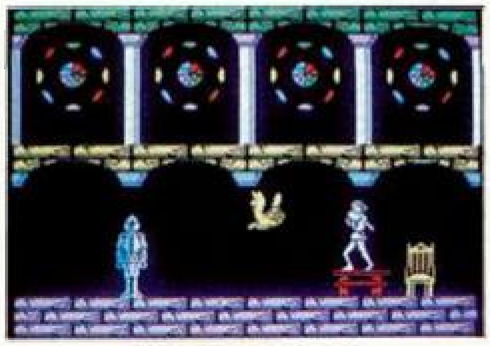
MICRO Manía

Sólo para adictos

CAMELOT WARRIORS

Parece mentira la habilidad y rapidez que poseen nuestros lectores a la hora de desvelar las interioridades de los programas. Hace apenas un mes que salió al mercado este magnífico, pero difícil programa, Camelot Warriors, y ya José Andrés Val nos envía desde Valladolid el mágico Poke para obtener las anheladas, deseadas y siempre bien recibidas «vidas infinitas».

Como siempre y por no variar, teclear MERGE"" y colocar antes del RANDOMIZE USR lo siguiente:



POKE 50782,255

Luego hacer RUN (ENTER) y poner en marcha el cassette. Lo demá es cosa vuestra...

THREE WEEKS IN PARADISE

En las últimas semanas os hemos ofrecido un emocionante serial microfónico en el cual el eminente escritor Dessas Trosso, nos contaba todas las claves para que el desamparado Wally consiguiera rescatar a Herbert y a su amada Wilma de las garras de los caníbales.

Pero como hemos podido comprobar por los millones de llamadas recibidas en nuestra redacción, el asunto de cómo conseguir hacer un agujero en el hielo, no había quedado excesivamente claro. Así pues nos vemos en la obligación moral de solventar estas pequeñas molestias que hayamos podido ocasionar.



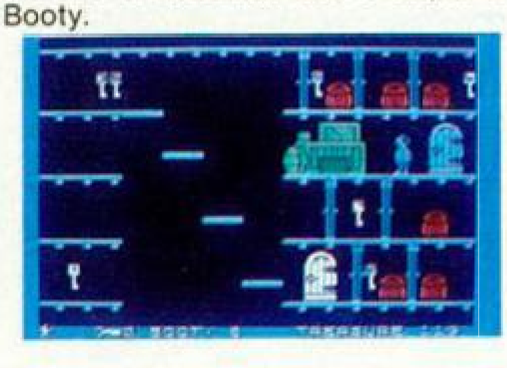
Exactamente esto es lo que tendréis que hacer: cuando lleguéis a la pantalla helada (la cual os será bastante fácil de reconocer porque está helada y que corresponde a la de la fotografía) situaros justo al lado de un agujero marcado que se encuentra en el suelo de la misma. Ahora, gracias a que previamente hemos cogido la menta y que en este momento la tenemos en nuestro poder, simplemente con empujar el joystick hacia adelante, la pantalla cambiará y aparecerá un hermoso y negro agujero. Lo cogemos pulsando la tecla correspondiente, y ya podremos seguir con el resto de la aventura. Perdón.

BOOTY

A continuación os ofrecemos un maravilloso programa cargador que os permitirá obtener un ilimitado número de vidas para disfrutar a tope de este antiguo pero estupendo arcade llamado Booty.

```
10 CLEAR 26870
20 LOAD"" SCREEN$
30 LOAD"" CODE 26880
40 RAND USR 26880
50 POKE 58294,0
60 RAND USR 52500
```

Para hacerlo funcionar, cargar este programa antes del juego original y posteriormente ejecutarlo y poner en marcha el cassette con la copia de Booty.



LIBROS



QUILL, TRATAMIENTO DE TEXTOS EN EL QL

F. Simón/C. Spottiswoode/Bluepint
Editorial Anaya. 188 páginas

El QL fue uno de los primeros ordenadores que incluyó, con la propia máquina, un paquete de Software básico.

Uno de los programas que integran el paquete es el procesador de textos QUILL, desarrollado por PSION.

Se trata en esta ocasión de un magnífico libro, de cuidada edición, que permite al usuario novel aprender con facilidad cómo se maneja dicho programa.

Lo más destacable de este volumen es, en nuestra opinión, que está dirigido a personas que carecen de experiencia anterior en el mundo de los microordenadores.

En el primer capítulo hay una toma de contacto con el QL y sus características diferenciales respecto de otros ordenadores.

Inmediatamente después pasa a analizar el QUILL con todo lujo de detalles, desde una pormenorizada descripción del formato de pantalla, hasta todos los aspectos referidos a las instrucciones, terminología, etc.

Durante todo el libro se hace gala de un gran sentido del humor, lo que le da un aire divertido y ameno, bastante necesario en esto de la tecnología informática.

Son muy de agradecer también los múltiples consejos de utilidad que se citan en todo momento.

También alude, de una forma bastante profusa, a todos los demás típicos de la edición de documentos, como son tipos de letra, márgenes de impresión, ajuste de texto, tabulaciones, búsqueda y sustitución de palabras, párrafos estandarizados, encabezamientos y pies de página, impresos, etc.

Por último, trata uno de los aspectos más oscuros de todo este mundillo: la instalación de la impresora, con un completo repertorio de recomendaciones específicas sobre impresoras atípicas.

TECLASTICK

Jesús Alonso RODRIGUEZ

Entre las cartas que recibimos de nuestros lectores una de las preguntas que nos plantean con más frecuencia se refiere a la forma de leer el teclado de un modo eficaz y sencillo. Por ello, hemos desarrollado una colección de rutinas en código máquina que estamos seguros, servirán para que nuestros lectores las incorporen en sus porpios programas.



Cuando el programador se plantea la realización de un juego, uno de los problemas que surgen es encontrar una forma eficaz de leer el teclado. El Basic confía todo el problema a la función INKEY$, pero este procedimiento tiene muchas limitaciones; la principal viene dada por la imposibilidad de leer más de una tecla cada vez, con lo que resulta imposible simular el funcionamiento de un joystick, no se pueden obtener las direcciones diagonales, ni moverse al tiempo que se dispara.

La solución a esto consiste en leer directamente los «pórts» que configuran el teclado, pero este método, además de resultar lento, mantiene algunos de los inconvenientes de la función INKEY$; por un lado, resulta complejo escribir programas que sean compatibles con joystick y, por otro lado, el teclado se lee sólo en un pequeño momento del bucle de juego que puede ser bastante largo; esto lleva a una cierta lentitud de respuesta en las teclas que se aprecia en algunos de los programas que nos envían nuestros lectores. No es buena práctica de progrmación el tener al programa esperando por la entrada del usuario, pero tampoco lo es el obligar a éste a estar pendiente de cuando el programa quiera leer la tecla que está pulsando. En nuestra coleccción de rutinas, hemos hecho uso de la interrupción enmascarable para solventar este problema.

Simulemos una función "Stick"

Sería útil contar con una función a la que pudiéramos llamar para que nos leyera el teclado como si fuera un joystick, es decir, atendiendo sólo a las cuatro teclas de «cursor» (las flechas) y a una que hiciera de disparador (por ejemplo: «ENTER») y nos devolviera un dato en función de la o las que hubiera pulsadas, de forma similar a como lo hace un joystick Kempston al leer en IN 223.

Supongamos que nuestra función se llamára «STICK», nuestro programa podría ser totalmente compatible con joystick, si quisiéramos usar el teclado, haríamos: «LET a=SITCK» y si fuera el joystick: «LET a=IN 223»; a partir de aquí, el programa no debería diferir en absoluto, ya que el dato entregado por ambas funciones sería el mismo.

Bien, lamentablemente el Spectrum no posee esta función, pero podemos simularla con el uso de la función «USR» y una pequeña rutina en código máquina. En la FIGURA-1 se muestra el listado en Assembler de una rutina que lee las teclas de cursor y la de «ENTER» y nos devuelve en «BC» el dato que nos daría un joystick tipo Kempston. Si el lenguaje Assembler fuerte no es tu fuerte, no te preocupes, el LISTADO-1 contiene el código máquina correspondiente a esta rutina, puedes cargarlo en memoria utilizando el Cargador Universal de Código Máquina publicado en nuestras páginas,

LISTADO 1

```
1   0100003EBFDBFECB4720  1033
2   02CBE13EF7DBFECB6720  1550
3   02CBC93EEFDBFECB5720  1502
4   02CBC1CB5F2002CBD9CB  1353
5   672002CBD10600C90000  756
```

LISTADO 2

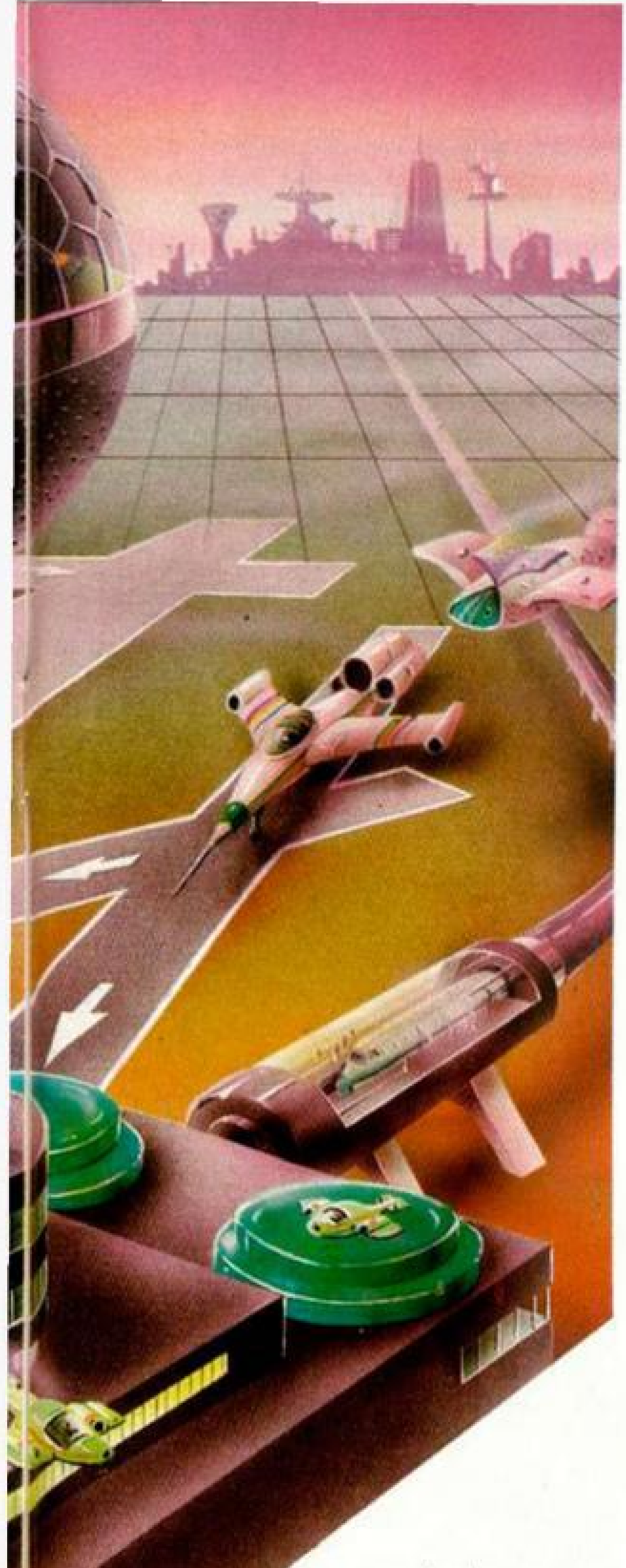
```
1   01EBFFF5C50100003EBF  1187
2   DBFECB472002CBE13EF7  1518
3   DBFECB672002CBC93EEF  1518
4   DBFECB572002CBC1CB5F  1491
5   2002CBD9CB672002CBD1  1206
6   79A7280332815CC1F1ED  1273
7   4D3EEAED47ED5EAF3281  1366
8   5CC93E3FED47ED56C900  1250
```

LISTADO 3

```
1   0100003E7FDBFECB5720  985
2   02CBE13EFBDBFECB4720  1522
3   02CBD93EFDDBFECB4720  1516
4   02CBD13EDFDBFECB4F20  1486
5   02CBC9CB472002CBC106  1116
6   00C90000000000000000  201
```

LISTADO 4

```
1   01EBFFF5C50100003E7F  1123
2   DBFECB572002CBE13EFB  1538
3   DBFECB472002CBD93EFD  1516
4   DBFECB472002CBD13EDF  1478
5   DBFECB4F2002CBC9CB47  1467
6   2002CBC179A728033281  940
7   5CC1F1ED4D3EEAED47ED  1681
8   5EAF32815CC93E3FED47  1174
9   ED56C9000000000000000  524
```



o cualquier otro que te construyas. Puedes colocar la rutina en cualquier dirección de memoria que te venga bien, ya que es reubicable.

Una vez que la tengas cargada, puedes utilizarla de la siguiente forma:

1. El código máquina deberá estar salvado en cinta a continuación del programa donde quieras utilizarlo.

2. La primera instrucción de tu programa deberá ser:

```
CLEAR d-1: LOAD «»CODE d
```

donde «d» es la dirección de memoria a partir de la que deseas cargar tu rutina. Para usuarios de 48 K, sugerimos que «d» puede valer 65315, y para usuarios de 16 K, puede ser 32547.

3. Cada vez que necesites leer el teclado, utiliza la instrucción:

```
LET a = USR d
```

donde «d» es, de nuevo, la dirección a partir de la cual colocaste la rutina. El dato que obtendrás en «a» estará en función de la tecla o teclas que hubiera pulsadas según se muestra en la Tabla 1.

Si tu ordenador tiene el antiguo teclado de goma, es muy posible que las teclas de cursor te resulten incomodísimas para mover; en ese caso, es preferible utilizar la «Q» para mover hacia arriba, la «A» para hacerlo hacia abajo, la «P» a la derecha y la «O» a la izquierda, asimismo, resulta cómodo utilizar la «M» como disparador. Si es este tu caso, te será más útil la rutina que mostramos en la FIGURA 3 y a la que corresponde el LISTADO 3. Esta rutina hace exactamente lo mismo que la anterior, pero tomando en consideración las teclas «Q», «A», «O», «P» y «M».

Máxima atención al usuario, 50 veces por segundo

En los programas de juego escritos en Basic ocurre, con frecuencia, que el bucle de juego es muy largo, hay que mover aleatoriamente muchos objetos antes de que el jugador pueda mover el suyo; supongamos que el bucle de juego dura 2 segundos (una eternidad en informática), en un punto del bucle haremos: «LET a = USR d» que leerá el movimiento del jugador; pero esta instrucción sólo se ejecutará una vez cada 2 segundos, si el jugador no está preparado en ese instane, habrá perdido su turno. En muchos programas ocurre esto, y resulta muy desagradable el que haya que tener oprimida durante 2 segundos una tecla para asegurarse de que el ordenador la ha leído.

Sabemos, por otro lado, que la ULA interrumpe el funcionamiento del ordenador cada 20 milisegundos para obligarle a que lea el teclado; pero esta lectura no nos sirve, ya que sólo es capaz de identificar una tecla de cada vez. La solución están en hacer que la interrupción fuerce al microprocesador a saltar a una dirección de memoria donde nosotros habremos colocado nuestra rutina de lectura del teclado, y nos almacene el dato en una posición de memoria donde podremos leerlo desde Basic cuando llegue su momento; la rutina deberá modificar el dato solamente si hay

Figura 1

```
      Nombre de la rutina: TECLADO-1
   Dirección de inicio: REUBICABLE
              Longitud: 48 bytes
Condiciones de entrada: NINGUNA
           Descripción: Lee las teclas de cursor y la de "ENTER"
                        como si se tratase de un joystick
                        KEMPSTON y devuelve en "BC" el dato
                        correspondiente a las que hubiera
                        pulsadas.

100         ORG   60000
110         LD    BC,0
120         LD    A,#BF
130         IN    A,(254)
140         BIT   0,A
150         JR    NZ,LB1
160         SET   4,C
170 LB1     LD    A,#F7
180         IN    A,(254)
190         BIT   4,A
200         JR    NZ,LB2
210         SET   1,C
220 LB2     LD    A,#EF
230         IN    A,(254)
240         BIT   2,A
250         JR    NZ,LB3
260         SET   0,C
270 LB3     BIT   3,A
280         JR    NZ,LB4
290         SET   3,C
300 LB4     BIT   4,A
310         JR    NZ,LB5
320         SET   2,C
330 LB5     LD    B,0
340         RET
```

Figura 2

```
      Nombre de la rutina: TECLADO-2
   Dirección de inicio: 60159
              Longitud: 79 bytes
Condiciones de entrada: Se ejecuta en cada interrupción. (20ms.)
                             ACTIVAR: USR 60220
                          DESACTIVAR: USR 60231
           Descripción: Lee las teclas de cursor y la de "ENTER"
                        como si se tratase de un joystick
                        KEMPSTON y almacena, en la dirección
                        23681, el dato correspondiente a las que
                        hubiera pulsadas.

100         ORG   60159
110         DEFW  60161
120         RST   #38
130         PUSH  AF
140         PUSH  BC
150         LD    BC,0
160         LD    A,#BF
170         IN    A,(254)
180         BIT   0,A
190         JR    NZ,LB1
200         SET   4,C
210 LB1     LD    A,#F7
220         IN    A,(254)
230         BIT   4,A
240         JR    NZ,LB2
250         SET   1,C
260 LB2     LD    A,#EF
270         IN    A,(254)
280         BIT   2,A
290         JR    NZ,LB3
300         SET   0,C
310 LB3     BIT   3,A
320         JR    NZ,LB4
330         SET   3,C
340 LB4     BIT   4,A
350         JR    NZ,LB5
360         SET   2,C
370 LB5     LD    A,C
380         AND   A
390         JR    Z,CERO
400         LD    (23681),A
410 CERO    POP   BC
420         POP   AF
430         RETI
440 ACT     LD    A,#EA
450         LD    I,A
460         IM    2
470         XOR   A
480         LD    (23681),A
490         RET
500 DES     LD    A,#3F
510         LD    I,A
520         IM    1
530         RET
```

Figura 3

```
      Nombre de la rutina: TECLADO-3
   Dirección de inicio: REUBICABLE
              Longitud: 52 bytes
Condiciones de entrada: NINGUNA
           Descripción: Lee las teclas "Q", "A", "O", "P" y "M"
                        como si se tratase de un joystick
                        KEMPSTON y devuelve en "BC" el dato
                        correspondiente a las que hubiera
                        pulsadas.

100         ORG   60000
110         LD    BC,0
120         LD    A,#7F
130         IN    A,(254)
140         BIT   2,A
150         JR    NZ,LB1
160         SET   4,C
170 LB1     LD    A,#FB
180         IN    A,(254)
190         BIT   0,A
200         JR    NZ,LB2
210         SET   3,C
220 LB2     LD    A,#FD
230         IN    A,(254)
240         BIT   0,A
250         JR    NZ,LB3
260         SET   2,C
270 LB3     LD    A,#DF
280         IN    A,(254)
290         BIT   1,A
300         JR    NZ,LB4
310         SET   1,C
320 LB4     BIT   0,A
330         JR    NZ,LB5
340         SET   0,C
350 LB5     LD    B,0
360         RET
```

alguna tecla pulsada, de lo contrario, deberá dejar el dato anterior.

De esta forma, el ordenador está atendiendo al usuario en todo momento y no se pierde ninguna pulsación; por otro lado, el programa en Basic dispone siempre de un dato actualizado sobre cuál ha sido la última orden que ha dado el jugador. La rutina que se muestra en la FIGURA-2 utiliza la dirección de memoria 23681 (una variable del sistema que no se utiliza) para almacenar este dato, y el programa en Basic deberá borrarlo cada vez que lo lea, de la siguiente forma:

```
LET a = PEEK 23681: POKE 23681, 0
```

La rutina se ejecuta cada vez que el microprocesador responde a una interrupción enmascarable, por lo que es necesario activarla cambiando el vector de interrupción y pasando a modo 2; esto se consigue con:

```
RANDOMIZE USR 60220
```

También es posible desactivarla con:

```
RANDOMIZE USR 60231
```

Mientras la rutina está activada, su existencia resulta absolutamente transparente para el sistema, de hecho, el usuario no tiene por qué darse cuenta de que está funcionando, salvo por el hecho de que, en todo momento, tendrá en la dirección 23681 un dato actualizado sobre la última tecla (o teclas) de cursor que se han pulsado.

El LISTADO-2 corresponde a esta rutina; no es, en absoluto, reubicable, ya que el vector de interrupción obliga a que se encuentre precisamente a partir de la dirección 60159, no obstante, si posees conocimientos suficientes de Assembler, podrás cambiar el vector de interrupción y dirigirlo a otro lugar de la memoria (si no posees los conocimientos, pero estás siguiendo nuestro curso, muy pronto podrás hacerlo).

Igual que en el caso anterior, para utilizar esta rutina en tus programas deberás hacerlo de la siguiente forma:

1. Salva el código máquina a continuación del programa en Basic donde quieras usarla.

2. La primera instrucción del programa deberá ser:

```
CLEAR 60158:load «» CODE 60159:
RANDOMIZE USR 60220
```

3. Cuando necesites el dato del teclado, ejecuta:

```
LET a = PEEK 23681: POKE 23681,0
```

Si omites la segunda parte (el POKE), podrás obtener un efecto de mantenimiento de la última orden que puede quedar muy bien en algunos juegos.

TABLA-1

MOVIMIENTO	SIN DISPARO	CON DISPARO
Sin pulsación.......:	0	16
Derecha.............:	1	17
Izquierda...........:	2	18
Abajo...............:	4	20
Arriba..............:	8	24
Derecha + Arriba....:	9	25
Derecha + Abajo.....:	5	21
Izquierda + Arriba..:	10	26
Izquierda + Abajo...:	6	22

Tabla 1: Datos entregados por las rutinas.

```
     Nombre de la rutina: TECLADO-4
    Dirección de inicio: 60159
               Longitud: 83 bytes
Condiciones de entrada: Se ejecuta en cada interrupción (20ms.)
                          ACTIVAR: USR 60224
                        DESACTIVAR: USR 60235
            Descripción: Lee las teclas "Q", "A", "O", "P" y "M"
                         como si se tratase de un joystick
                         KEMPSTON y almacena, en la dirección
                         23681, el dato correspondiente a las que
                         hubiera pulsadas si es que hay alguna,
                         de lo contrario, no hace nada.

100       ORG   60159
110       DEFW  60161
120       RST   #38
130       PUSH  AF
140       PUSH  BC
150       LD    BC,0
160       LD    A,#7F
170       IN    A,(254)
180       BIT   2,A
190       JR    NZ,LB1
200       SET   4,C
210 LB1   LD    A,#FB
220       IN    A,(254)
230       BIT   0,A
240       JR    NZ,LB2
250       SET   3,C
260 LB2   LD    A,#FD
270       IN    A,(254)
280       BIT   0,A
290       JR    NZ,LB3
300       SET   2,C
310 LB3   LD    A,#DF
320       IN    A,(254)
330       BIT   1,A
340       JR    NZ,LB4
350       SET   1,C
360 LB4   BIT   0,A
370       JR    NZ,LB5
380       SET   0,C
390 LB5   LD    A,C
400       AND   A
410       JR    Z,CERO
420       LD    (23681),A
430 CERO  POP   BC
440       POP   AF
450       RETI
460 ACT   LD    A,#EA
470       LD    I,A
480       IM    2
490       XOR   A
500       LD    (23681),A
510       RET
520 DES   LD    A,#3F
530       LD    I,A
540       IM    1
550       RET
```

Figura 4

La rutina de la FIGURA-4 y el LISTADO-4 sirven para aquéllos que tengan el teclado de goma y prefieran trabajar con las teclas «Q», «A», «O», «P» y «M» en vez de las del cursor. En este caso, la activación se hace con USR 60224 y la desactivación con USR 60235.

El funcionamiento de la rutina es sumamente sencillo. El vector de página de interrupción es «EAh», por lo que el verdadero vector de interrupción se lee desde «EAFFh» es decir, 60159d. En esta dirección y la siguiente está el número 60161 que es la dirección a donde salta el microprocesador. Primero se hace un «RST» #38» para leer el teclado e incrementar FRAMES, de modo que la rutina sea transparente al sistema. Después se preservan en la pila los registros «AF» y «BC», se leen los «ports» correspondientes a las teclas que nos interesan y se va componiendo en dato sobre el registro «C». Finalmente, se comprueba si el dato es «cero», y si no lo es, se almacena en la dirección 23681 antes de recuperar los registros que se preservaron y retornar desde interrupción con «RETI».

En la subrutina de activación (ACT), se carga en «I» el vector de página «EAh», se selecciona el modo 2 de interrupción y se pone a «0» el contenido de la dirección 23681. Al desactivar (DES), se coloca en «I» el vector original «3Fh» y se selecciona, de nuevo, el modo 1 de interrupción.

Esperamos que estas rutinas sirvan para mejorar aún más la calidad de los programas de nuestros lectores, y estamos seguros de que las usarán incluso aquéllos que no comprendan todavía el lenguaje Assembler.

Microfichas y LOGO

Para un programa que estoy haciendo, necesito más de 21 GDU (25 ó 26 exactamente). ¿Hay algún método para conseguirlo?

¿Es posible cargar las rutinas de las microfichas con el cargador universal de código máquina publicado en el número 31 de esa misma revista?

Soy un entusiasta del LOGO y les quedaría agradecido si me dicen el nombre de algún programa para Spectrum con el que pueda programar en este lenguaje.

Angel LUENGOS - Bilbao

☐ En el número 42 de nuestra revista, explicábamos la forma de obtener más de 21 GDU en el Spectrum. La explicación es demasiado larga para reproducirla aquí, pero el procedimiento es sencillo.

Para cargar las rutinas de las microfichas, hay que utilizar el cargador especial que apareció en la microficha «R-0», o uno que usted se haga al efecto. No se puede utilizar el cargador universal de código máquina del número 31 porque el formato del código fuente es distinto.

Existe un magnífico «LOGO» de la firma «Logo Computer Sistem. Inc.» Para mayor información, puede dirigirse a:

Logo Spectrum Club de Consultas.
Provenza, 281.
08037 Barcelona.
Teléfono: (93) 215 83 37.

Un fallo del sistema

Quisiera que me dijesen por qué al añadir 15 BORDER 0 al programa número 6 del artículo «Movimiento con teclado y joystick (I)» publicado en la revista número 33, el programa no funciona, y cómo se puede solucionar.

También me gustaría saber cómo colocar caracteres sueltos con su propio color de tinta y papel en cualquier columna de las dos últimas líneas (la 22 y 23) desde un programa. Ya sé que se hace con PRINT #1 ó #0 pero esto implica que se borre toda la línea hasta la columna donde se quiere poner el carácter y no hace caso al INK o PAPER que se pongan después de PRINT.

Ramón GALLASTEGUI - Guipúzcoa

☐ Efectivamente, parece que alterar el color del borde tiene efecto sobre los ports del teclado. Indudablemente, constituye un fallo del sistema, aunque puede corregirse si a cada dato de las líneas 120 y 130 se le suma 64.

Respecto a su segunda pregunta, no entendemos a qué se refiere. En teoría, la instrucción PRINT debe funcionar, por el canal «K», exactamente igual que por el «S». Nosotros lo hemos probado con la siguiente línea:

```
PRINT #; AT 0,0; «*»; AT 0,10; PAPER 1; «*»: PAUSE 0
```

Y funciona correctamente, realizando la impresión en la forma que cabría esperar.

«Disco ROM»

Tengo la intención de construir el «Disco ROM» que publicaron en los números 58 y 59 de la revista y me encuentro con el problema de que en las tiendas de electrónica e informática de mi localidad no disponen del conector para el «Slot» posterior del Spectrum. Les agradecería me indicaran si me lo pueden enviar ustedes contra reembolso o darme la dirección de la casa que lo suministra o de algún comercio en Madrid donde pueda encontrarlo.

David ORDOÑEZ - Asturias

☐ Lamentamos no poder remitirle el componente que nos pide, pero no somos distribuidores de material electrónico. Con frecuencia, nos escriben lectores solicitándonos material de hardware, interfaces, etc. Aprovechamos la ocasión para aclarar que, en estos casos, deberán dirigirse a los distribuidores, cuyas direcciones encontrarán en las páginas publicitarias de nuestra revista. Nosotros, lo más que podemos suministrarles son las placas de circuito impreso para nuestros montajes, a través de PROHOBBY.

En su caso concreto, no es necesario que el conector tenga exactamente la longitud del «Slot». Normalmente, son más largos y hay que cortarlos. No obstante, hay tiendas (al menos en Madrid) que los venden cortados a la medida.

«ZX 81»

Somos muchos los que hemos pasado al Spectrum desde el ZX-81 que, desde entonces, yace en un rincón; surgiéndonos, a veces, el pensamiento de si con una chapuza de «hard» o de «soft» no podríamos conectarlo al Spectrum para que nos sirviera de RAM, de ROM o de lo que fuera. Y, en general, qué podríamos hacer con el viejo ZX-81. ¿Podéis darnos un consejo?

Juan ESTADA - Madrid

☐ Es difícil pensar alguna aplicación para el ZX-81 en conexión con un Spectrum. En principio, los problemas surgen con la propia conexión. Habría que desarrollar un interface especial, ya que el ZX-81 no dispone de RS-232 ni nada que se le parezca. Por otro lado, necesitaría otro televisor o monitor para el ZX-81.

Respecto a las posibles aplicaciones, podría utilizarse como unidad temporal de almacenamiento (si es de 16K, claro) o establecer una pequeña red de área local para alguna aplicación que requiera el uso de dos ordenadores (el ejemplo más típico podría ser el juego de los «barquitos»).

En cualquier caso, los problemas de conexión parecen difíciles de resolver si no se tienen unos amplios conocimientos de electrónica digital y habría que diseñar un software que se encargase de gestionar las comunicaciones. Si se anima a hacerlo, no deje de contarnos los resultados. Quizá interese a otros lectores.

«DRAW» en código máquina

Estoy realizando un programa en el cual necesito utilizar muchos «PLOT« y «DRAW» por lo que he decidido hacer una subrutina en c/m para solucionarlo, pero no sé qué registros deben contener las coordenadas del «DRAW» negativo.

Alberto DE LA PUENTE - Madrid

☐ En la rutina del comando «PLOT», hay dos puntos de entrada, si se entra por 22DCh (8924), las coordenadas han de estar en la pila del calculador; pero si se entra por 22DFh (8927), las coordenadas han de estar en el registro «BC», concretamente, «y» en «B» y «x» en «C».

El caso del comando «DRAW» es más complicado, ya que puede trazar líneas curvas mediante aproximación de rectas. Suponemos, no obstante, que lo

que a usted le interesa es trazar líneas rectas, por lo que puede usar la rutina «DRAW-LINE» de la ROM. Esta rutina traza una recta desde las coordenadas que se encuentren en la variable «COORDS» (dirección 23677 y 23678, «x» e «y» respectivamente), hasta unas coordenadas que vendrán dadas por «x + dx» e «y + dy» donde «dx» y «dy» son los desplazamientos positivos o negativos (de la misma forma que en Basic).

De nuevo, la rutina «DRAW-LINE» tiene dos puntos de entrada; si se entra por 24B7h (9399), los desplazamientos deben estar, con su signo, en la parte alta de la pila del calculador; si, en cambio, se entra por 24BAh (9402), los desplazamientos deben estar de la siguiente forma: ABS (dy) en «B», ABS (dx) en «C», SGN (dy) en «D» y SGN (dx) en «E».

Mezcla de vídeo

¿Hay alguna forma de mezclar (superponer) imágenes de ordenador y televisión para ser grabadas en vídeo? ¿Existe en el mercado algún mezclador de este tipo?

Manuel HERNANDEZ - Málaga

☐ Varios lectores nos han hecho ya esta pregunta.

La mezcla de señales de vídeo es, considerablemente, más difícil que en el caso de sonido. El motivo es que las señales a mezclar deben estar sincronizadas, es decir, tener coincidentes los impulsos que determinan el inicio de cada línea y de cada cuadro.

Existen en el mercado mezcladores de vídeo, pero ignoramos cómo resolverían el problema del sincronismo en el caso de un Spectrum, ni siquiera, si ello sería posible. Tal vez, sea más fácil grabar la imagen del Spectrum en un vídeo-cassette y realizar la mezcla desde él.

Anti «BREAK»

¿Cómo puedo hacer para que si pulso la tecla «BREAK» cuando se esté ejecutando un programa, en vez de pararse o borrarse la memoria, saltase a la línea 1?

Adolfo MARTIN - Madrid

☐ La rutina «ON ERROR GO TO» publicada en la MICROFICHA R-1 le servirá perfectamente para evitar que el programa se detenga con un informe de error que no sea «0 OK», «STOP Statement» o «End of file»; por tanto, tampoco lo parará la pulsación de «BREAK». Tenga en cuenta, no obstante, que si tiene enchufado un Interface 1, esta rutina no funcionará con los mensajes de error propios de este dispositivo que sí detendrán el programa.

Es sencillo modificar esta rutina para que sólo actúe con el mensaje «BREAK into program» y no con los demás: Elimine las líneas 220 a 270 y añada las siguientes:

```
220 CP #15
239 JR NZ, CONT
```

E inicialice la rutina con: RANDOMIZE 1 = USR dirección. Esto hará que se salte a la línea 1 al producirse el mensaje: «L BREAK into program» mientras que los demás mensajes serán tratados de la forma habitual.

DE OCASION

● VENDO para Spectrum 16/48 K, Plus, sintetizador de voz Currah Micro-Speech. Mejorando al ordenador enviando el sonido por el televisor. Regalo manual en castellano e inglés y cinta de demostración en castellano. Interesados llamar al tel.: (958) 27 01 67 de 9,30 a 11 noche. Preguntar por Carlos. Precio: 8.500 ptas. Dirección: J. Carlos Martín. C/ Pedro A. de Alarcón, 87, 2.° Granada.

● VENDO monitor Zenith fósforo ocre comprado en junio 85 por 19.000 ptas. Spectrum Plus con mando Quickshot II y revistas por sólo 21.000 ptas. Atari computer Sistem con 4 mandos por 16.000 ptas. Envíos contrareembolso. Alexis Gutiérrez. C/ Gutiérrez Rada, 2. Laredo (Cantabria). Tel.: (942) 60 62 25.

● VENDO ordenador ZX-81 con ampliación de memoria de 16 K, en perfecto estado. Precio: 10.000 ptas. Dirigirse a José Isaac Aranda Jiménez. C/ Condesa Pardo Bazán, 22. Barcelona 08027. Tel.: 349 90 82.

● COMPRO periféricos para el Spectrum en buen estado. Interesados escribir a: Jesús Pérez Sicilia. Avda. Carlos III, 39, 2.° 3.ª. Córdoba 14014.

● VENDO Interface programable de Indescomp, con embalaje original e instrucciones. También puedo vender Joystick Sony 55 (Hit-Bit). José Martínez Coll. C/ P. Coronación, 13, 1.°. Onteniente (Valencia). Tel.: (96) 248 18 81.

● VENDO Spectrum 48 K, con cables, fuente de alimentación y cinta de demostración por sólo 31.000 ptas. Además regalo Interface para Joystick. Interesados contactar con Michael Lepoutre. Camino de Hoyarrasa, 67. Alcobendas (Madrid). Tel.: (91) 650 10 46.

● VENDO Zx-48, con 24 revistas, 5 libros, interface Kempston y un póster del ordenador Plus, todo esto por sólo 50.000 ptas., a convenir. Llamar al tel.: 661 64 83. Sant Boi (Barcelona). Preguntar por José M.ª Oliver.

● DESEARIA ponerme en contacto con usuarios del Spectrum para el intercambio de ideas, información sobre el tema etc. Interesados llamar al tel.: (94) 447 80 83. Bilbao. Preguntar por Alfonso.

● VENDO números atrasados de MICROHOBBY por 90 ptas., (sin el curso de Basic y Código Máquina). Gastos de envío incluidos, poseo los n.° del 1 al 46. Interesados dirigirse a Fermín Bernaus. C/ Juan de Garay, 5, 3.° A. San Sebastián 20008.

● VENDO Spectrum 48 K más interface Kempston con sonido, 2 libros y 30 revistas. Todo por 36.000 ptas. Escribir a Andrés Moya. C/ Pí y Margall, 139, 5.° C. Vigo 2, (Pontevedra). Tel.: (986) 29 04 58.

● INTERCAMBIO mapas. Tengo los siguientes: «Dum Darach», «Pyjamarama», «Gyron», y así 15 más. También tengo pokes (80). Interesados contactar con Roberto. C/ Travesía de Vigo, 192, 4.° A. Vigo 7. Pontevedra.

● BUSCO usuarios del Spectrum de 48 K y MSX, para el intercambio de utilidades. Compro impresora barata. Escribir a Isidro Bernabéu Feliú. Apartado 11. Jijona (Alicante).

● VENDO Oric-Atmos, con manuales y cable Centronics. Interesados llamar al tel.: (93) 330 42 57 (21-22 h.).

● VENDO Spectrum 48 K, por 23.000 ptas. 50 revistas nacionales e inglesas por el precio de 4.000 ptas. Todo 33.000 ptas. O bien lo cambiaría por un Amstrad. Interesados pueden llamanr al tel.: (91) 705 92 87. Preguntar por Carlos.

● VENDO Zx Spectrum Plus de 64 K, completamente nuevo, aún en garantía, con todos sus accesorios (fuente de alimentación, cables, cinta de demostración, etc.). Y con dos manuales en castellano e inglés, Interface I de Sinclair y un Microdrive con varios cartuchos, 6 libros sobre el Spectrum, revistas, precio en el mercado alrededor de 120.000 ptas. Precio especial, junto o separado. Enviar ofertas a J. M. Moreno. Avda. Gral. Elizaicin, 9, 4.°. Alicante. Tel.: (965) 20 35 40.

● VENDO ordenador Zx Spectrum 48 K, con cables, fuente de alimentación, un manual para aprender su manejo y un libro de Basic, cinta Horizontes, garantía. Pago gastos de envío. Todo por 25.000 ptas. Urge venderlo. Interesados llamar al tel.: (91) 798 25 33. Preguntar por Javier Fdez. Morato. Madrid. Escribir a J. J. Martínez Seco, 47, 2.° Dcha. Madrid 28021.

● VENDO lápiz óptico de la casa DKT a estrenar con manual de instrucciones Interface y cassette de demostración con programa de dibujo de 16 y 48 K. Precio: 4.000 ptas. (negociables). Escribir a José Antonio Rodríguez Ovalle. Apdo. correos, 28. Villafranca del Bierzo (León).

● INTERCAMBIO de ideas, trucos, información para el ordenador Spectrum 48 K. Interesados escribir a Félix López Jordán. Pza. Fuensanta, 2, 13.° B. Murcia 30008.

● VENDO circuitos integrados para ordenadores y uso generalizado: lógicos, memorias, micros, etc. Mando lista gratis. Interesados dirigirse a Miguel Angel Clavijo. José Antonio, 5, 1.° Izqda. La Cuesta (Sta. Cruz de Tenerife).

● VENDO amplificador de sonido para Spectrum, con o sin altavoz, no precisa conectarse al bus del ordenador. Precio a convenir. Escribir o llamar a Fco. José Cazalilla, C/ San Roque, 78. Castellón. Tel.: (964) 22 78 26.

● VENDO Spectrum 48 K, en buen estado, gran cantidad de revistas. Precio: 25.000 ptas. O bien lo cambio por un Commodore 64. Interesados llamar al tel.: (91) 797 24 03. Preguntar por Javier.

● CAMBIO las instrucciones del HOBBIT en español por las de HULK, Pyston u otras, sin importar idioma. Escribir a Ignacio Cabrera Peset. Pje. Residencias Luz, 14. Valencia 46010.

● VENDO amplificador Optimus ART 30 wat., salidas exteriores de hasta 500 ohm., entradas para plato, platina, etc., con columna de diodos, etc. Su precio actual es de 60.000 ptas., lo ofrezco por la mitad (30.000 ptas.). Interesados llamar al tel.: (981) 31 68 69 y preguntar por Carlos.

● VENDO teclado Saga-1 (65 teclas), impecables por 8.000 ptas. Impresora Seikosha GP-50S a estrenar por 15.000, con 4 rollos de papel. VENDO/CAMBIO por 1 Microdrive o un Interface Kempston y Joystick, un Commodore VIC-20, instrucciones en castellano y un libro, en perfecto estado de funcionamiento por sólo 10.000. CAMBIO sintetizador de voz monofónico Yahama CS-01-II, con instrucciones por 1 Interface I más 1 Microdrive (precio 35.000 ptas.). Interesados dirigirse a Luis Torrents Nadal. Tel.: (96) 333 73 08. Valencia.

● VENDO, por cambio de ordenador las revistas de MICROHOBBY del n.° 1 al 49, por el precio de 4.500 ptas., regalando el curso de Basic encuadernado. Interesados llamar al tel./: 693 98 72. Madrid. Preguntar por José L. Liébana Pastor.

● VENDO Spectrum 48 K, casi nuevo, con garantía Investrónica, en perfecto estado, con fuente de alimentación, cables, manual en castellano y cinta de demostración. Además regalo teclado profesional DKT, revistas del tema por sólo 30.000 ptas. Escribir a Jorge Bellido Merino. C/ Baluarte, 68, 3.°. Barcelona 08003, o bien llamar al tel.: (93) 310 17 34 a partir de las 22 h.

● VENDO Joystick tipo Kemspton e Interface programables Stonechip, con muy poco uso, por sólo 7.500 ptas. Interesados llamar al tel.: (928) 25 29 67. Antonio.

● VENDO impresora GP-50S, para Spectrum, papel normal. Precio: 17.000 ptas. También impresora GP-100 AS, interface RS-232, papel 10", tracción, 80 columnas, 50 c.p.s. Funciona con el interface 1. Interesados llamar al tel.: (91) 218 04 12. Preguntar por Rafael (hijo). Precio: 25.000 ptas.

● COMPRO un ZX-81 estropeado, daría como máximo 3.000 ptas. o usado (7.000 ptas.) con todos los accesorios: cables, fuente de alimentación y manual. Interesados llamar al tel.: (954) 16 17 49 (4 a 8 tarde). Preguntar por Federico. Sevilla.

● COMPRO impresora GP-50 de Seikosha que esté en buen estado. Javier Roldán C/ Arturo Soria, 37. Madrid. Tel.: 408 97 67.

● DESEARIA intercambiar todo tipo de información para Spectrum 48K y Commodore 64K. Interesados escribir a Gonzalo López Infante. Vasco Núñez de Balboa, 3, 7.° D. Huelva 21004. Tel.: (955) 25 59 23.

● VENDO ordenador Amstrad CPC-664 con unidad de disco 3" incorpora monitor color 14". Comprado en Oct-85, con garantía oficial. Precio mínimo 98.000 ptas. al contado (aceptaré otras ofertas). Urge venta. A. Moreno. C/ Riera Alta, 43 At-3.°. Barcelona 08001.



PERIFERICOS SPECTRUM A POSTERIORI

El **Spectrum** se rodea siempre de lo mejor.
Si, a priori, te decidiste por el más popular de los ordenadores, decídete, a posteriori, por los más importantes periféricos.

Interface 1
Permite la conexión de hasta ocho Microdrives, que pueden guardar hasta 680KB de datos y programas.
Tiene un canal RS232 para conectar impresoras, modems, ...
A través de su toma de red de área local puede comunicarse con otros 63 **Spectrum.**

Microdrive ZX
Para almacenar más de 85KB en cada cartucho magnético, con un tiempo de acceso de 3,5 segundos.

Interface 2
Para usar programas contenidos en cartuchos ROM. Contiene un controlador para dos Investicks.

Investick
Joystick de gran robustez, con mando ergonómico, ventosas para fijarlo y cuatro puntos de disparo, para disfrutar más con los juegos.

Interface programable con sonido
Permite usar los Investicks en **todos** los programas, con los efectos sonoros al volumen deseado.

Lápiz óptico
Para crear directamente en la pantalla todos los dibujos imaginables. Se puede dibujar, colorear, reducir/ampliar, mezclar con textos u otros dibujos, grabar y cargar pantallas. ...

SPECTRUM A PRIORI

DISTRIBUIDOR EXCLUSIVO **investronica**

Tomás Bretón, 62. Tel. (91) 467 82 10. Telex 23399 IYCO E. 28045 Madrid
Camp, 80. Tels. (93) 211 26 58 - 211 27 54. 08022 Barcelona

etc